

KEYFRAME

MAGAZINE

THE PUBLICATION FOR LIGHTWAVE, INSPIRE, AND AURA ENTHUSIASTS

Deep Blue Sea

Bitten by the Shark



**“Leafing” Trees
Creating Plug-in GUIs
Boat Wakes**

Place with computer magazines



Issue #14 Sept/Oct 1999
\$9.95 US \$14.95 Canada

Hollywood on y

PARTICLE STORM



Storm of the Century images courtesy of Netter Digital Entertainment, Inc. Storm effects added with Particle Storm 2 Pro.



The Hunley image courtesy of Station X Studios, LLC. Explosion created with Particle Storm 2 Pro; water created with NatureFX.

AVAILABLE FROM



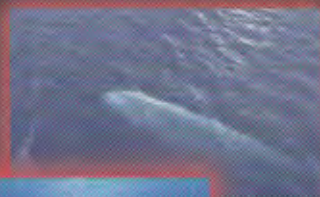
800-544-6599 • 414-548-8120

WWW.SHARBOR.COM

The most popular LightWave plug-in in the world!

Advanced features include: Particle Fields, Hair, Spawner, Sub-Objects, Sub-Polys, Mimic, Assembler, Envelopes, Layout Integration, Sticky Particles, Bounding Shapes, Collision Detection, Collision Avoidance, OpenGL Realtime Preview, Gravity, Wind, Flocking & Swarming, Drag, Force Fields, Death Wish, Particle Shader, and Parenting!

your Desktop...



Suntory images courtesy of Euphonic, Inc.
Water effects created with NatureFX.

Realism Unparalleled...

The software responsible for brining the Titanic back to life is now available for LightWave! NatureFX includes these advanced features: NatureFX Clouds for creating 3D volumetric clouds, NatureFX Air to simulate the coloration of the sun, horizon, and sky, and to generate haze and rainbows. NatureFX Ocean creates waves, wakes and ripples, and NatureFX Object Bob gives you bobbing objects.

CREATED BY



800-944-5520 • 414-549-2200
WWW.DYNAMIC-REALITIES.COM

KEYFRAME

MAGAZINE

The Editor

Doug Kelly dakelly@earthlink.net

Advertising Rep

Tiffani Banaszak keyframe1@aol.com

Art Director

Tiffani Banaszak keyframe1@aol.com

Co-Conspirators, Writers, and Artists

Pat Beck zippie@execpc.com

Jared Bendis jaredjared@earthlink.net

Jason Goodman jason@msd-nyc.com

Robert Gougher www.robougher.com

Julian Kain julian@juliankain.com

Graham McKenna

graham@grahamprismatic.free-online.co.uk

Matt Morgan violator@thethreshold.net

Steve Pugh steve@foundation-i.com

Lee Stranahan lee@stranahan.com

Phil South phil_south@hotmail.com

David Warner dwarner@albany.net

Jeremy Yates jeremy@thethreshold.net

The Digital Underground

DMG Publishing

2763 West Avenue L, Suite 162

Lancaster, CA 93536 USA

Phone 805-264-4991

Fax 805-264-4998

How To Get It

888-778-WAVE

805-264-4991

<http://www.keyframemag.com>

Keyframe Magazine is published 6 times annually by DMG Publishing. © 1998 DMG Publishing. All rights reserved. Reproduction, in part or in whole, without written permission is strictly prohibited. Printed in the USA.

The Secret Formula

Keyframe Magazine is created on Intergraph machines using Quark 4.0, Photoshop 4.0, LightWave 5.6, Painter 5, and various plug-ins. Hardware includes a MusTek 2400 9600dpi scanner, Iomega Zip and Jaz drives, and a DPS Perception. The cover CD is created by David Warner using a GateWay2000 PPro200 w/128 mb Ram and an 18GB hard drive, a Truevision Bravado 1000 video capture card, Sony CD-Recorder, and an Iomega Zip drive. Software includes LightWave 5.6, Aura, Photoshop 4.0, Macromedia Director 6.5, Adobe Premiere 4.2, QuickTime 3Pro, and Adaptec Easy CD Creator.

No dongles were harmed in the making of this publication.

Departments

3 Editorial

by Doug Kelly

- Very Interesting Times

4 Ear to the Wall

- Artbeats New Titles
- Jangaroo Videos Available
- Project: messiah Revealed at SIGGRAPH
- Foundation Imaging Receives Emmy Award
- New ToasterNT
- Deep Paint
- Marlin Studios' Amazing Sci-Fi Textures
- Space Wars Digital Effects Kit

12 Macintosh Up Close and Personal

by Jason Goodman

- LightWave Laptop on the Set

14 On Topic

by David Warner

- SIGGRAPH Threads, Align Ending, Making QTs

17 SIGGRAPH Pictorial

by Jared Bendis

- Some moments to remember

24 Cover Story

by Bill Mesa

- A look at *Deep Blue Sea*

40 Reviews

- LipService by Doug Kelly
- Amorphium by Jared Bendis
- Graphics Chipsets by Julian Kain
- *Masters of Visual Effects* by Lee Stranahan

48 Final Cut

by Steve Pugh

- Foundation Imaging's Workflow Pipeline for RSST

Tutorials

8 Character Shop

by Doug Kelly

- An interview with FT's Jeff Scheetz

18 On A Wing and A Prayer

by Graham McKenna

- Winged humanoid character setup

21 Batadactyl

by Jeremy Yates

- Character and scene setup

22 Leaf it to Me!

by Matt Morgan

- Morphing and texture-mapping leaves

28 Building Image Processing Plug-ins

by Rob Gougher

- Creating GUIs

32 Heavy Metal

by Phil South

- Surfacing metals

36 Five Minute Miracle

by Pat Beck

- Faking a boat wake

EDITORIAL

VERY INTERESTING TIMES...



This issue marks my one-year anniversary as Editor. It's been an interesting year. We've tried to raise the mastery level of our tutorials, to print incisive reviews of products you can use, and to put the industry news in perspective with more in-depth analysis. Judging from our subscription statistics, Internet buzz, and direct feedback from some of you, we're doing a good job. For that, I thank our contributors for creating so much good material, and the staff and management at DMG Publishing for their hard work and unfailing support.

I want to continue improving *Keyframe*. To do that, I need to hear from you. Which features of this magazine do you find most useful? Which do you pass over? What do you want more of? What are we missing? Email me directly, post your views to the LightWave Mailing List or comp.graphics.apps.lightwave, or drop a note in the mail. I read it all, and I respond to it personally. Those who have contacted me in the past year can attest that *Keyframe* is the better for their suggestions.

The next year would be a challenge even if we were simply aiming to preserve the status quo. However, I believe that the old curse, "May you live in interesting times" is only a bad thing if you're not prepared to ride the shockwaves. There are a few good ones coming up. NewTek is releasing LightWave 6.0, which is going to dramatically change the way many of us model, animate and render. Be prepared to unlearn, then relearn most of what you think you know about 3D. Aura 2.0 is going to open your eyes to the possibilities of painting and compositing as an integrated part of the NewTek suite. Rounding out the production process, VideoToasterNT will change how you import, edit and export video footage. Third-party software and plug-ins such as Station X Studios Project: messiah, FILMBOX matchmove, and Dynamic Realities NatureFX will bring even more feature film power to your desktop. You are going to be able to kiss goodbye most of your time-consuming, compromise-forcing workarounds and put that effort into creating better art.

With all these new tools, you're going to need reviews to help you choose the ones you need. You're going to want tutorials to explain how to use the new tools in ways the manuals don't hint at and the programmers never dreamed of. You're going to want case studies from the best artists who can inspire you to greater creativity. You're going to want programming tips on coding your own custom tools. You're going to want real-world advice from top working professionals to help you build your career. That's what the next year of *Keyframe* is going to bring you.

May we all continue to live in interesting times!

Doug Kelly

Ear to the Wall

News from the LightWave arena

farmland, forests, mountains, deserts and more. Spectacular vistas of snow-covered peaks, sandy beaches, bustling cities and other scenes are also included.

Each title contains royalty-free footage in D1 resolution and sourced from motion picture film. The clips are delivered in Apple QuickTime format, encoded with PhotoJPEG at a high quality setting. The compression is not designed for playback directly from the CD, rather it is an archival codec offering the best compromise of file size and superior image quality. The titles are available in NTSC or PAL format for Macintosh, Windows, and Unix users. In addition, a conversion utility to output a sequence of Targa frames is included for Intel-based Windows users whose applications do not support QuickTime. Users who want to purchase individual clips can do so from the Artbeats Web-site at www.artbeats.com.

Animated Surface Maps contains motion imagery that is not only seamless vertically and horizontally but also seamlessly loopable in time. The imagery was adapted from footage taken from the Artbeats Digital Film Library. Each product contains QuickTime movie files that are 400 x 400 pixels in size, 30 frames per second and average 9 seconds in length. They can be used in any 3D modeling or rendering application that can import a QuickTime movie. Each clip is a seamless loop in time and a seamless tile in space.



- **Animated Surface Maps Volume 1-Ambience** contains 32 background textures that evolve or "morph" over time. Much of the imagery was taken from high speed film footage of liquid which gives an impression of flowing fabric or shiny oil. The textures were designed to be used as backgrounds, environments, displacement maps and more.
- **Animated Surface Maps Volume 2- Fire** contains 15 pyrotechnic textures including fire, smoke and fireworks. Due to the nature of fire, 11 of the clips are not seamless vertically.
- **Animated Surface Maps Volume 3- Water** contains 24 textures of bubbles, white water, caustic patterns, and generic surfaces taken from film footage of lakes and pools.

Visit www.artbeats.com for more information on these and other CD collections.



Artbeats Explodes with New Titles

Myrtle Creek, OR - August 10, 1999 - Artbeats Software, Inc. announced that they have added two new royalty free stock footage products to the Artbeats Digital Film Library: ReelExplosions 4 and Aerial Landscapes, as well as a set of three motion imagery shaders titled Animated Surface Maps.

Reel Explosions 4 is the latest addition to the ReelExplosions series. Like RE3 (ReelExplosions 3), RE4 contains large scale zero-gravity explosions, many containing debris. Most of the explosions come with a close-up version as an added clip. The closeups were made during the telecine process and since the source was 35mm film, there was no loss of resolution or "softening" during enlargement. The user gets more than 50 clips for the price of 30 because of the additional closeups. As in RE3, the clips were shot with a high speed 35mm camera at 360 fps. Most of the clips include mattes that can be used as alpha channels for easy compositing.

Aerial Landscapes gives the user 30 picturesque aerial clips of

Jingaroo Children's Videos Now Available as Three-Part Series



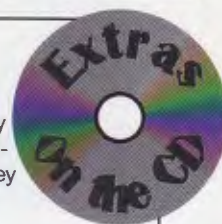
The second and third video tapes of the new three-part children's series, "Jingaroo," were recently completed by Beckett Entertainment of Dallas. The video series, which uses LightWave 3D, is geared toward children ages 2-11 and features the adventures of a soccer-playing kangaroo and his unique crew of Outback friends. Together they explore and discover the exotic three-dimensional world down under, and meet many new friends along the way.

Jingaroo Volume 1 recently won a 1999 Telly Award for Entertainment from the Center for Creativity. The titles are: Jingaroo Volume 1: "Water Tree and Other Stories;" Jingaroo Volume 2: "The Fool Moon and Other Stories;" and Jingaroo Volume 3: "Outback Grand Prix and Other Stories." They retail for \$14.95 each at www.jingaroo.com.

This series is the first project produced by the new entertainment division of Beckett Publications, headed by executive producer CR Conant. Beckett Entertainment's mission is to product quality family entertainment that tells fantastic stories with positive themes. Initial distribution of the tapes and merchandise will be through mail-order catalogues and the web site. Beckett Entertainment will produce more direct-to-video tapes and is currently shopping the pilots for a national television series.

Conant worked with Jingaroo's co-creator Mark Stokes, as well as over twenty artists using LightWave 3D from Dallas-based DNA Productions. Beginning with concept drawings, the computer artists translated the two-dimensional drawings into three-dimensional life. "With Jingaroo, we are trying to visualize a world that children will find unique and magical," says Keith Alcorn, co-founder of DNA. Menza Production Group of Dallas is composing all the original music and sound design for the series. "We've written original children's pop tunes and infused them with world music elements to create a very unique sound that kids enjoy," says Menza. Original songs from the first series will be released on CD later this year.

"This project signifies a new era in sophisticated children's programming," says Conant. "By combining classical artistic craftsmanship and blending it with cutting-edge animation technology, we've created a dynamic look that will appeal to every member of the family."



Project: messiah Revealed at Siggraph



SANTA MONICA, Calif. - Aug. 24, 1999 - In what may be a special effects industry first, Station X Studios announced that it will license the production version of its proprietary animation software tools, code named "Project: messiah," as a commercial software product. The high-end animation production software, developed to animate the studio's forthcoming live-action/CGI feature film "Dungeons & Dragons: The Movie," can be used on computers utilizing the Microsoft Windows NT operating system with either Intel compatible or Alpha compatible processors. A Mac version is also in the works.

In a move designed to achieve maximum distribution, Station X is introducing the product for a limited time at a special price of \$395 per seat.

"Project: messiah has five years of artist wish lists designed right into it; it's fast, powerful and easy to use. One of the most important, and often overlooked, advantages of messiah is that it's developed and tested in a production environment, yielding a superior design between the way artists want to work and the fast-changing world of effects production," stated software pioneer Fori Owurowa, president of Station X. "While it may seem counter-intuitive to sell proprietary technology to your direct competitors, becoming a significant software company was always part of our business strategy. Besides, Station X still has exceptional creative and technical in-house assets to help us keep our edge in digital animation and effects."



Owurowa indicated that the decision to release this powerful animation software into the market was also prompted by the desire of its feature film clients for Station X to be able to collaborate with other effects studios. "By making this software widely available, we also feel that we will significantly increase the qualified digital artist pool from which we can draw, in order to ramp up for large-scale projects on short notice," added Owurowa.

Station X Software Developers Lyle Milton and Dan Milling, making up the messiah development team along with Owurowa, stated, "Project: messiah was designed from the ground up to help revolutionize the art of 3D character animation. Not only does it have features found only in packages costing thousands of dollars more, but also those not found in any other package."

Some of the advanced features of Project: messiah include: IK; bones; expressions; easy character setup; real-time interactivity; local/world coordinates on-the-fly; forward/inverse kinematics; procedural/key-frame animation blending; and interactive soft-body dynamics. Effects can modify multiple objects so users can animate one skeleton and let it operate on an entire army.

Designed for Windows 95/98 or NT Intel or Alpha CPU, the package's system requirements also include 8mb of available disk space, 32mb of RAM, CD-ROM drive, and NewTek's LightWave3D or Inspire. For more information, visit Station X's Web site at www.stationxstudios.com.



Foundation Imaging



VALENCIA, CA - August 28, 1999 - Foundation Imaging, already winners of an Emmy award for their visual effects work on the pilot for *Babylon 5*, added another Emmy to their list of awards Saturday when *Star Trek: Voyager's* episode "Dark Frontier" won for Outstanding Achievement in Visual Effects for a Series. Taking home statues were CGI Supervisors Rob Bonchune and Adam 'Mojo' Leibowitz, as well as Foundation's Director of Animation John Teska.

Foundation Imaging would also like to recognize the artists and technicians who contributed to this episode's success: Composers Sherry Hitch and Pam Vick, CG Artists John Allardice, Koji Kuramura, Brandon MacDougall, David Morton, Trevor Peirce, Kyle Toucher, and Dan Ritchie, Coordinator Lindsay Adler, and Systems Engineers Michael Donahue, Jeremy Lang, and Jason Meier.

Star Trek: Voyager • Dark Frontier • UPN • Paramount Pictures

Dan Curry, Visual Effects Producer/Supervisor; Ronald B. Moore, Mitch Suskin, Visual Effects Supervisors; Elizabeth Castro, Arthur J. Codron, Visual Effects Coordinators; Paul Hill, Don Greenberg, Visual Effects Composers; Greg Rainoff, Visual Effects Artist; Rob Bonchune, CGI Supervisor; MOJO, CGI Supervisor; John Teska, CGI Animator.

The Way Emmys are Made

Toast for Dessert



San Antonio - August 25, 1999- NewTek announced that the Video Toaster for Windows is now shipping to dealers worldwide at a SRP of \$2995.00. The new Video Toaster is the heir to the original Video Toaster for the Amiga, which initiated the desktop video revolution in 1990.

The Video Toaster offers users a suite of tools for video creation including all the essential components for video editing, compositing and 2D/3D animation. This uncompressed D1 (Composite, Component, Y/C), Windows NT solution provides a level of price/performance unmatched in the industry. Unique to the Video Toaster is a breakthrough technology developed by NewTek that turns a standard Pentium II or III processor into a highly specialized video DSP (digital signal processor) for time base correction and other capabilities typically associated with expensive custom chipsets.

"By using an extremely efficient, proprietary assembly language code, we are able to use the computer's processor to emulate tens of thousands of dollars of specialized video hardware," said Tim Jenison, founder and CEO of NewTek. "Because of this, the Video Toaster is superior to sophisticated systems that cost tens of thousands of dollars."

The Video Toaster, which includes a version of LightWave 3D animation software, Aura paint and compositing software and in-sync's Speed Razor 4.5 SE video editing software, requires no proprietary drives. Complete systems capable of editing up to an hour of video can be assembled for as little as \$8000.00.

Uncompressed Video, No Compromise

The Video Toaster provides an uncompressed D1 (Composite, Component, Y/C) digital video solution typically associated with extremely expensive video systems. High-end facilities know that uncompressed digital video is an essential component to the video editing process. Most compression-based systems have been relegated to the industrial and corporate video segments due to restrictions to the number of layers and quality degradation associated with multi-pass compression. All compressed systems produce unwanted artifacts. Because Video Toaster utilizes uncompressed video, users can infinitely layer video without any concern for these artifacts. This is currently a major separation between high-end and mid range systems. The Video Toaster brings that ultra high-end sophistication to users at all levels at an unprecedented price point.



Powerful Editing Capabilities

Video Toaster includes one of the most powerful editing applications in the industry: in-sync's Speed Razor 4.5 SE. This editing application offers:

- a project timeline that lets users easily create professional video projects containing unlimited layers of video, audio, still image files and animation
- the ability to capture, edit and output to uncompressed video and other video formats, including ASF for video streaming on the Web and multimedia AVIs
- comprehensive 2D compositing system
- graphical workspace for digital media content management
- real-time on-screen preview of video edits
- power character generator (CG)
- In addition to Speed Razor SE capabilities, the Video Toaster offers a waveform monitor, vectorscope and real-time 60 field onscreen preview.

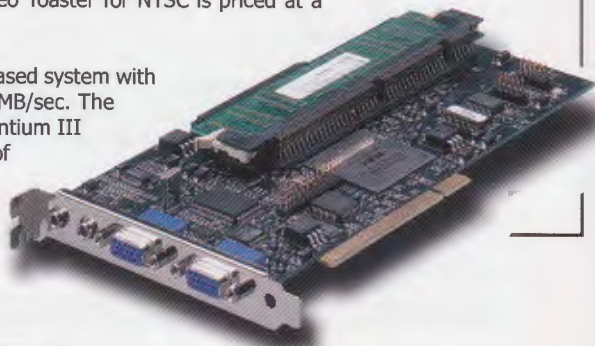
A Toaster Built to Last

The Video Toaster was designed to last for years to come. Similar to the original Video Toaster for Amiga, which is still in production 10 years after its launch, the Video Toaster utilizes a hardware system that leverages highly tuned software and the system's processor. The true power of the system comes from the proprietary software designed to create magic without the use of expensive video cards or bulky external hardware. By merging software and hardware, the Video Toaster can be upgraded and expanded well beyond current systems. NewTek will be adding modules for a real time 16 input switcher, storyboard editing, real time 3D DVE, chroma keying, serial D1 interface, and other components to be announced.

Price and Availability

Video Toaster is now available at NewTek authorized dealers worldwide. Video Toaster for NTSC is priced at a SRP of \$2995.00. NTSC and PAL (dual format) solution retails for \$3495.00.

Minimum system requirements for running Video Toaster are a Windows NT-based system with a Pentium 266 processor, 128 MB of RAM and a drive system capable of 22 MB/sec. The recommended system configuration is a Windows NT-based system with a Pentium III processor, 256 MB of RAM and a drive system capable of 22 MB/sec. A list of tested and approved computer systems is available at <http://www.newtek.com>.



You're In Deep Paint



Los Angeles, CA - August 10, 1998 - Right Hemisphere™ announced the

availability of Deep Paint 3D™ at Siggraph. Deep Paint 3D provides designers with authentic artistic media - such as oils, watercolors, crayons and pastels - that can be "brushed" directly onto 3D models. With a single stroke Deep Paint 3D paints complete material or shader definitions, and can be used to easily brush surface textures, such as skins or woods, directly onto objects. It offers full support for 3D Studio Max 2.x and 3.0 and also a bi-directional shader interface to Photoshop. In addition, it supports a range of complementary products, including Photoshop plug-ins.

"The shipping version of Deep Paint 3D builds on the strengths of its predecessor, 4D Paint™, the market leading paint tool for desktop 3D art and animation. With its new name, Deep Paint 3D gains powerful new paint features, performance enhancements, a simplified interface, and full support for 3D Studio Max 2.x and 3.0. It also offers a bi-directional shader interface to Photoshop and support for a range of complementary products, including Photoshop plug-ins. These enhancements strengthen Deep Paint 3D's position as the most complete Windows-based 3D-paint product available.

"Deep Paint 3D is well on its way to becoming the standard in 3D Paint," said Greg Punchatz, director of animation at Janimation. "Deep Paint 3D allows me to use incredibly authentic paint tools and to apply textures, like various skins, to my 3D designs, with an interface that is straightforward to use. Powerful real-time rendering technology makes it possible to see the subtleties of my work in progress. We have been using the product for years, and we see it making our projects more and more rewarding and satisfying."

Pricing and Availability

The Windows Intel version of Deep Paint 3D is completing extensive customer testing now, and will be available from Right Hemisphere in August 1998 for \$785. The Alpha processor version is scheduled for a Q4 1998 release and will be \$995. Deep Paint 3D requires Pentium or Alpha processors running Windows NT or 98, with 64 MB of RAM and 30 MB of hard disk space, and 800 x 600 x 16 bit color graphics. Visit www.righthemisphere.com for more information.

That's a Wrap!

Graphics studio and publisher Marlin Studios' CD "Amazing Sci-Fi Textures" contains over 550mb of textures that can be used to create high-quality science fiction and space scenes. The product is the latest in the company's series of photorealistic texture collections created for use by 2D and 3D graphic artists. Previous releases include "Seamless Textures You Can Really Use," "Seamless Textures2- Rustic Exterior Surfaces," and "Seamless Textures3- Ultimate Interior Surfaces."

Amazing Sci-Fi Textures features texture maps and materials created by award-winning graphics artist and sci-fi buff Geoff Holman, who also created the Seamless 2 Exteriors collection. The unique textures are great for 3D art, games, film CGI work, high-end web design, print media, commercial video, visualizations, or any application requiring the depiction of science fiction and space scenes. The patterns are varied enough to be used as textures for other applications, especially the large assortment of interesting ship hull textures, which feature fascinating metal plating patterns. The textures also lend themselves to many uses in 2D graphics programs, such as background patterns in Photoshop.

The textures on the CD can be used on Intel, Alpha, Mac, and Amiga platforms. Amazing Sci-Fi Textures retails for \$149. Visit Marlin Studio's web site at www.marlinstudios.com.



Space Wars: A New Frontier



Animators of all levels now have an affordable, all-in-one solution for creating eye-popping space battles and spectacular stellar animations. The Space Wars Digital Effects Kit by Digital Soundstage is a new CD-ROM that brings together all the elements needed to create fantastic space animations. This collection includes 3D objects, texture images, explosion and fire image sequences, background plates and even sound effects! Every element is royalty-free.

All of the elements from the Space Wars DEK were created by experienced Hollywood professionals. 3D animation company Digital Soundstage created all of the 3D models and texture images. Hollywood effects veteran Robbie Knott created the explosion and fire sequences, which are from the Artrights ReelExplosions and ReelFire series. The sound effects were created by Hollywood sound artist John Peters are from his "The Hollywood Edge Science Fiction Sound Effects Library."

The Space Wars DEK works with any program that accepts LightWave or 3D Studio object formats, and is available for Intel, Alpha, Mac, and Amiga on one CD. A special introductory price of \$129 is being offered. For more information contact Digital Soundstage at 818-563-3600 or visit them at www.digitalsoundstage.com.





Get A Job: An Interview with Jeff Scheetz

I recently had the opportunity to sit down with Jeff Scheetz, Project Lead of Foundation Imaging's teams for *Starship Troopers*. Jeff is responsible for (among other things) reviewing demo reels and conducting interviews. If you want to work at Foundation, you'll need to see Jeff. He provided some valuable (and sometimes surprising) advice to, and comments on, the current LightWave talent pool.

DK: *Are you seeing an improvement in the quality of demo reels that you review?*

JS: No, in fact I'm seeing a sharp decline.

DK: *Really? Would you assign any reasons to that?*

JS: I think that all the really good people are working. The reels I'm getting don't make my eyes bleed or anything, but you caught me in the middle of my big disappointment festival. [*Foundation Imaging*

had been recruiting heavily, and on the day we interviewed Jeff he was doing open walk-in interviews in conjunction with SIGGRAPH '99. - Ed.] We're just not getting as many good submissions as we once were. In the early days of LightWave, there were all these people who learned at home and didn't know how good they were. They kept getting better and better while taking lesser jobs and honing their craft. One day they would hit Hollywood with this really stellar reel that got them their first job. Recently the demand has gotten so great that those types of people were being pulled out and given jobs at a much earlier level in their development. Right now there are too many people who bought the software last week, and are trying to get out there and sell themselves.

DK: *Are you seeing the LightWave tutorials or demos on these people's reels?*

JS: Sometimes, but not very often. Occasionally I get a person whose reel looks like they just rendered whatever was on the LightWave CD-ROM. I've had people send me the little space fighter animation, which is especially funny here, since Ron [Thornton, founder of Foundation Imaging] did it.

DK: *That's the classic plagiarist's blunder, handing the piece to the person who actually did it.*

JS: Yeah, that happens to other people here too. Some of our

people work freelance or have come from other projects, and I'll see a piece they did show up on an applicant's demo reel. And I'll ask our people about it, and they'll tell me the real story. Once we had somebody send us, essentially, a whole other company's demo reel and try to pass it off as their own work. I guess he thought that just because this other company was in Texas I'd never figure this out. It turns out that the owner of this other company is a good friend of mine, and (laughing) it came back to haunt this person in about an hour. I have also shared this story with many of my friends at other companies.

DK: *I warn people at every opportunity that CGI animation is a very small community. I was at the CG-CHAR meeting earlier this week, and there were fewer than 600 people in the room, and someone said that this was probably most of the character animation talent on the planet. It only takes one person standing up in a group like that, saying "So and so is trying to pass my work off as their own" to kill your career.*

JS: That'll kill you.

DK: *What advice do you have for someone who has LightWave, thinks they're hot stuff, and wants to apply here? What do you want to see, what do you not want to see on their demo reel?*

JS: It's not what I don't want to see, as much as what I do want to see. Even if it's stuff that people do a lot of (space-ships are a good example), you should



Jeff Scheetz photo courtesy of Jared Bendis

do what you really want to do, and have the guts to just do that. A lot of people will try to show that they can use so many different parts of the program, or handle so many animation genres, that they haven't really taken the time to master any one of them. If there's a part that they really enjoy, they should really focus on that, learn it well, and then they'll have something to offer the community.

When we hired for *Troopers*, no matter what your specialty was, I could probably give you a job doing something very similar. We found so many people who had tons of very different stuff, but none of it was really good, and I couldn't hire them. If they're a jack of all trades and master of none, then I don't know what to do with them the first week, and that makes it really really hard to get them into the flow and have a starting point of usefulness.

DK: *So should people put their best work, what they really want to do, on their demo reel first...*

JS: Yeah.

DK: *...or only?*

JS: Well, I like to see some examples of something that has been completed, something that's not just another animation test. That's because finishing things is a big part of what we do as well.

DK: *You're not working to your own creative impulses, you're working to the client's.*

JS: Yeah. And it's never as good when I'll get a tape with just a head, talking, and the applicant will say, "I just wanted to show you that I could do lip sync." And another shot is a faceless character moving, and they say "I wanted to show you I could do a walk cycle." And instead of a cool set, these tests are done against the default gradients. Well, if you have a head saying something cool, and put that on top of a body that's doing a good walk cycle, in an interesting environment, then you have something! This one shot can be three times more powerful than it was because you were just trying to show me one thing.

DK: *So you want at least one finished piece, pulling it all together?*

JS: Yeah. I would hire someone based on a whole bunch of interesting tests, if I needed the kinds of individual skills they had proven. But by creating great shots, the whole is definitely greater than the sum of its parts.

DK: *So if you want to show lighting, still lifes are OK, if you want to show character animation, the wireframe skeletons are OK to show the timing and sense of weight, but you want to see at least one finished piece where it's all put together.*

JS: I just think it's a waste if you do a bunch of independent pieces that together would have been better. When we go through the *Trooper* process, we do a lot of mocap but still a lot of it is hand-keyed: Then we go through and do the face. Then we go through and do the eyes. Then we put it in an environment. Then we do lighting on it. And these are all individual skill sets that are pretty much required.

DK: *Does one of your animators do all that through an entire shot? It's not like, one just does lighting and all the shots come through on the production pipeline?*

JS: It varies a little bit, but everyone can do these things. And more often than not, they wind up taking a shot all the way through it. And they do this in a day! I mean, if someone needs to spend three weeks building something, I totally get that. But when they actually have all their production elements lined up, they should be able to create shots in a day or two, because that's really all the time they get here. [See *Steve Pugh's Final Cut* column in this issue for more details on the *Starship Troopers* workflow.-Ed.]

During interviews beginning animators will say, "I think I spent about a month on this." That really means about two hours on a Sunday, two half hours in the course of a weeknight, and then an entire weekend finishing it up, all over the course of a month. But if you can train yourself to sit in your chair and actually finish your shots, and know when they're done, it's much more advantageous than this nebulous "I've been tweaking this

walk cycle for five months" claim.

DK: *You feel pretty strongly about this.*

JS: I'm the project supervisor for *Troopers*, but I still hire for all our shows. I had to put together 70 people to come work on this show. And this week, I'm trying to find ten more. And it's really hard.

DK: *Can you hire people from Canada?*

JS: If they can work in the United States under the North American Free Trade Agreement it's not really a problem. NAFTA requires a related university degree. Most other types of visa can be tough. When *Troopers* started four months ago I hired Dave Jarrard, and yesterday they called me up to say that he can come into the U.S. It's worth it, because he's good, but it's not practical by any stretch of the imagination to wait that long while you are in production. It would have been more practical to find an American woman to marry him. (Laughs) Not that I would ever do that!

DK: *Let's get back to the demo reels for a moment. How soon do you get poisoned when you watch a reel?*

JS: I'll fast-forward before I'll stop watching. I always fast-forward to the end.

DK: *Are they there with you when you watch the reels?*

JS: Today they are, most of the time they're not.

DK: *Do you fast-forward with them here?*

JS: No, I'll sit through the whole thing. I always try to turn this around, to tell them what we're looking for. The most common problem new animators have is that they fall in love with their own work without comparing it to anything. I know it's really tough in a vacuum, because everyone you know is going to say, "Damn, that's really good! That's better than *Babylon 5*!" because they know you did it. But it's important to have the objectiveness to look at it fairly.

I suggest to people that they download a picture of Voyager, and paste

in their spaceship like it's in the same shot, then ask yourself if this picture makes sense. And I can't see where they'd actually say, "Yeah, it's just as good," even though it's three boxes with procedural noise on them.

DK: *Do you actually get that?*

JS: Yeah.

DK: *Oh, ow.*

JS: But everybody that applies to me gets my full, undivided attention, even at fast forward. Every day, mail comes when I'm at lunch, and the first thing I do when I get back is look at demo reels. Even if I've got a lot of other things to do, I do it first, because the most important thing I can do is to have a really good crew. And we do have a really good crew here. But we have a team on this show that from day one has run at least one man down. And then I've been able to replace that person, except then I'll need to create a specialty team and pull from different teams, you know, beg, borrow from so-and-so, whatever. So I have not been able to have a perfectly full crew yet.

DK: *You're dealing with company commander problems; you've always got attrition and every time you've got the special teams out, that's pulling your best troopers from their individual platoons.*

JS: But with all the unemployed animators, I'm still running with holes in the teams because I'm having trouble finding people who are good enough. And I'm not that big a snob.

DK: *They have to be able to pull a performance. That's what it comes down to. And that takes "practice, practice, practice."*

JS: Uh-huh. And the thing about it is, you should know. Because I remember staring at my work, before I had a job, and I was going, "What's not right?" And then I'd read Mojo's VTU articles. He had a really good "Here's how to do Babylon 5 shots in ten easy steps" article. I did everything it said and it

came out really good. I also tried to do other shots using the theory and spirit of what he was talking about.

DK: *Not just follow the recipe.*

JS: Not just follow the recipe, but definitely respecting the recipe, and not second-guessing it. He hired me. I watched him watch my reel, and it was like, mm-hm, mm-hm, mm-hm, *hired*, mm-hm, mm-hm, mm-hm, *hired*, mm-hm, mm-hm, mm-hm, *hired*, it was like that. Looking back I guess he was thinking, "He did what I said, and his job is to do what I say, so he's hired." That was pretty cool.

DK: *I'm active on the CG-CHAR list, and I know there have been some online classes and things like that. What do you think of the Internet as a critique and mutual support tool for animators out in the boonies? Is this a good thing?*

JS: It's good, but nobody likes to get rude.

DK: *Do you read the CG-CHAR list? I mean, they're not rude, but they're merciless.*

JS: Are they?

DK: *Oh yeah. They'll say things like, "there was no sense of weight, it just floated all over, your f-curves must be crap, you must have like three keyframes in thirty seconds" and they'll say that.*

JS: Then that's a *great* thing. Wow. What we're looking at, though, is increasing this company to double again, probably in the next year. What we're pretty much acknowledging is that's going to mean a sizable chunk of our build-up is going to be a school. We're still working out how we're going to do it.

DK: *You're going to have to ramp up teaching staff.*

JS: Yeah. There are teachers teaching LightWave, making money teaching LightWave, that I won't hire, because they're not very good. So how good are their students going to be?

DK: *Did you know that the enrollments*

in just about every LightWave school have dropped off drastically? I don't know if it's because people have decided they're not getting enough out of it or what.

JS: They're not. The schools come to us and ask us what we want, and we tell them, and they don't listen. I could totally write you a curriculum, tell you exactly what skills they need to work here. If they could do that, it would be a no-brainer to hire their students.

I totally believe that large scale CGI action-adventure shows aimed at eight-year-old boys is a big part of the future of LightWave. Because after *Troopers* comes out, kids are really not going to want to watch *Godzilla: The Animated Series* done in 2D. And if we can get enough people who will be content to do this kind of work, there will be enough work for everybody. And dare I say, we'll keep these jobs in America instead of seeing them go overseas. It's an excellent field to get into if you can find the right training, or you can train yourself intelligently. Maybe what we need is a curriculum on how to teach yourself LightWave. Do I know how to do the following things? Check this off before sending a reel to Hollywood.

DK: *Do you know how to do certain things; do you know how to evaluate and explore the use of a new tool?*

JS: Mm-hm. And do you know how to tape *Voyager* and watch animation shots sixteen times until you see what's really happening? Do you know how to keep this from getting too difficult? Do you know how to do something that you can actually finish in one day that's demo-reel quality? If you left me, or anyone at Foundation, alone with a computer, a PVR card, and a copy of LightWave, in a week (minus rendering time) I could totally do a demo reel that would get me a job where I need to be, because I know this stuff. And I think a lot of new animators think, "Well, I did my first walk cycle, so I'm ready to send out my demo reel." I hear that, "This is my first giz file, and this is my first explosion," and I think, show me your tenth one and maybe it'll be in better shape.

DK: *Yeah. I get email from people who buy my book, and ask "Where does the 3D stuff start, I'm bogged down in the storyboard stuff and don't get it."*

JS: Yeah, that's the other thing, just having the patience to sit down and practice. Practice! Dare we say practice!? They just think, I just have to get the software to prove I can do it.

One thing that's kind of interesting about this, is that age really has nothing to do with it. There's a guy here, Brandon MacDougall, who started with us a couple years ago, he's nearly forty, he's a model builder and he's really, really good: And his son Tom, who was, like, 19, was playing with Light-Wave and really getting into building characters. And his dad told him, "Hey, they're going to need this Cyclops creature for *Mystic Knights*." And a week later he shows up with it. And he's out of high school, so we hired him. Now he's head of the *Starship Troopers* character development team. And the funny thing is, he's got a fourteen-year-old brother who I would hire like that <snap> if it weren't for child labor laws. Right now he's just building pod racers and stuff like that for fun. But talent-wise, he's awesome.

DK: *This whole industry is very much a meritocracy, except where you've got the politics at the higher levels of the bigger studios. Do you care at all about what someone has on paper, or is it just what you see on the demo reels?*

JS: Just the demo reel. Because for so many people here, this is their first job in the entertainment industry, let alone the animation industry. So I don't really care about the paper. In fact, it would take an amazing resume to get me to call without seeing a tape. People will write on resumes "call for a demo reel" I think, what are you saving it for? If you want to work in Hollywood on TV or Movies there are only about 7 places to apply!

DK: *I warn people, this is like sending out a resume. You don't grudge a stamp or the paper or the envelope. You make a lot of 10-minute dubs and you mail*

them out to everybody and his kid brother, and then you follow up. But you have to do a stack of them. But they say, "It's a lot of money!"

How many post-hires, how many people either crack or don't work up to the level of their demo reel, they don't work well in teams?

JS: Almost none. A while back we had one guy who didn't really know Light-Wave when he started, which would have been OK, but he spent most of his time trying to convince everyone else we were using the wrong software. If you take on a job that requires using new software you should be ready to embrace it.

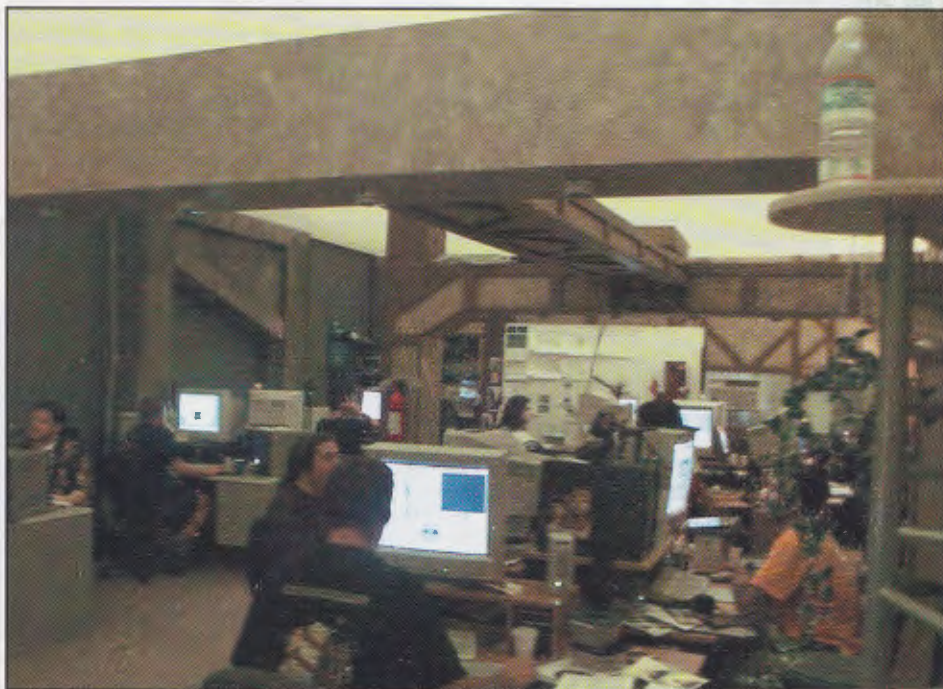
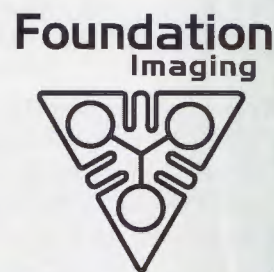
DK: *Have you had any hires that just didn't fit into the corporate culture?*

JS: There's not really much of a corporate culture. Foundation Imaging is full of very fun and sometimes VERY strange people, but as long as they keep doing great work and don't prevent others from working, I don't care how odd they are. That's also why my interviews are a breeze and I often hire over the phone. If I you want to work here, I like your work, and I have an opening you, will probably get the job! Why do we even need to meet?

You also need to be a team player. But here it wouldn't even occur to you to not be a team player, because there's so much of that around you. Everyone's swimming downstream, you don't think, hey, I'm going to turn around and swim upstream. A prima donna attitude would not last a minute here. No one dares even try it because no one else is getting away with it. Outside work it can be a little different. You catch a Foundation artist at a party full of wannabes, and he's can be quite a prima donna. You get a Foundation artist in a room full of other Foundation artists, and no one cares, it's just our job.

Send your demo reels (if you dare) to:

Jeff Scheetz
Foundation Imaging
24933 West Avenue Stanford
Valencia CA 91355



Modelers at work in The Foundry at Foundation Imaging. photo courtesy of Jared Bendis



Up Close and Personal by Jason Goodman

In last issue's column, I postulated about all the ways in which a laptop computer could be useful to the LightWave artist in the field. This month I tested my own suggestions on location during an actual shoot. Something that had not occurred to me, but which became very clear in the execution of this idea, was the invaluable added benefit of impressing the client. By doing something that our client had never seen done before, we really gave a wonderful impression. This is exceedingly important in today's 'what have you done for me lately' business world.

MSD animator Stephen Baker and I went on location with a PowerBook G3 300 and an Irez CapSure PCMCIA video capture card. The goal of this shoot was to create an element for a larger animation, which required several newscasters to appear on a wall of over sixty LightWave-generated television sets. The PowerBook proved useful on the bluescreen set, Figure 1. It assisted us in pinpointing the subtle details of the live action setup, and in recreating them in

our virtual set. The results were great and the whole procedure was surprisingly easy to set up and work through.

The key to this process is the video capture hardware in your laptop. I do not know of a laptop with built-in analog video capture capabilities (although it may exist) so the first thing you need to do is find a suitable third party solution. Most laptops offer expansion capabilities via PCMCIA or what is now referred to as PC Card or Cardbus slots. On many laptops, one of the slots features a capability known as Zoomed Video. This is often used for MPEG or DVD decoding cards if your laptop has a DVD player. The same technology that allows these high bandwidth devices to blast video onto your laptop screen in this case allows your laptop to capture video. If you are using one of Apple's new 400mhz or 333mhz G3 PowerBooks, or an iBook, you have the added option of using a USB video input device.

Well-known G3 upgrade manufacturer XLR8 currently offers the innovative InterView, a USB video I/O product. The InterView looks like a funky USB cable with RCA and S-Video connectors on it. One end plugs into your laptop's USB port and the other plugs into the source camera, VCR, TV, or whatever. It comes with all the software you need to capture footage, and even comes with Strata VideoShop, an aging but functional QuickTime editing and effects application. I briefly mentioned the InterView last month, but had fairly limited information about it. Although I still have not

tested it, I recently had an opportunity to see it in action at MacWorld New York, and I was impressed. InterView is currently available for Mac OS only! Fancy that. This is cool because it lets Mac users know that there still are manufacturers out there that are dedicated to the platform. It also usually means that you can expect better tech support as the technicians are not catering to the Windows majority. Despite the popularity of Mac LightWave, even our beloved NewTek is guilty of having a paucity of Mac-savvy technicians.

Before you throw away your Media 100, keep in mind that the InterView and products like it are not 'broadcast quality' video capture systems. The reason I opted to put that last statement in quotes is twofold. First of all, it is a statement that is thrown around by every manufacturer that wants you to buy their video product (whatever it might be) and secondly, it does not really imply anything of any significance. If you watch the news or shows like *America's Funniest Home Videos*, you may see people broadcasting low-grade VHS-C footage. Does that mean a \$199 camcorder from Best Buy is 'broadcast quality'? I suppose it all depends on the quality of your broadcast, but my point is that you need to take certain statements with a grain of salt. Devices that are truly up to professional standards will use more factual, feature-derived descriptive statements in their marketing and less superlatives.

All this having been said, image quality is not my primary fascination with



Figure 1: LightWave laptop on bluescreen set, operated by Stephen Baker.



Figure 2: Lighting setup.

the InterView, CapSure or other products like them. Video engineers and waveform monitor aficionados need not apply here. While these devices offer generally good quality, I personally would not use them for a professional video production of any importance. I also do not recommend using consumer oriented video products for creating or capturing footage that will eventually be compressed for delivery in low bandwidth mediums like the Internet or CD-ROM. In this example, the footage is used merely for reference within the LightWave scene. Many users may find these devices suitable for a wider range of applications, but you will have to make that determination.

All that being said, you should understand that the devices described here are way more than adequate for the intended bluescreen shoot. All we really need is a still frame. You may wonder why I would not grab this with a digital camera, and there is a good reason. It is important that the still frame or clip that we capture is coming directly out of the source camera. I want to capture subtleties in the shot such as perspective created by the focal length of the camera's lens and other elements that would not be reproduced in an image captured by a separate camera.

The primary advantage of the InterView over the Capsure is its ability to capture 640x480 and "up to 30fps". That is according to its marketing literature. This is deceptive however. The device will capture up to 30fps only in 320x240 mode. 640x480 is available only for stills. Again, it is not perfect. I have not qualified any of XLR8's claims, but I was told that capture frame rate is heavily dependent on the speed of your processor and resolution of your capture. For our needs, this specification is basically irrelevant. Since all we need is a still, capture frame rate is inconsequential. The image quality I saw at MacWorld was really nice. I have seen the InterView advertised for only \$89, so I would recommend USB equipped LightWave commandos who do compositing in the field look into it. The 300mhz machine I used for this shoot does not have USB, which is why I opted for a Card Bus (PCMCIA) Irez CapSure card. It cost \$129 and captured only at

320x240. I have since seen this same card advertised for \$99, undoubtedly due to competition from XLR8. While the CapSure offers decidedly lower resolution than the InterView it still handily met our requirements for this shoot. With new devices coming to market I would anticipate the possibility of further price reduction for this product, so it may remain a good solution for the budget conscious. Remember that this captured footage is only used as a reference for live action objects and lighting. The 320x240 resolution provided by the CapSure proved to be totally suitable.

Both devices also capture audio via the built-in microphone port on the G3 laptop, but again that was unnecessary in this case. Until LightWave 6.0 (with its audio import capabilities) ships, I cannot image a situation where you would use audio capture in conjunction with LightWave on location. Irez seemed more confident in the CapSure's ability to do 30fps, so if you find yourself in a situation that requires this frame rate, it may be a good choice for this reason as well. This card also deceptively claims that it will capture 640x480, but this is merely an interpolation of its native 320x240 frame size. Incidentally Irez was showing a CapSure Pro product at MacWorld that may be out by the time you read this. Its key benefit over the card I used is its ability to capture 640x480 full resolution frames. It also offered a small breakout box that featured S-Video input. For a more complex shot, the added quality could conceivably be a benefit. My determination to use the CapSure was more of a practical decision based on the expansion options my machine provided rather than the features of competing products. I do not recommend either of these devices as a video professional's primary digital I/O interface, so you may find yourself in a similar situation.

To create our element, we shot an actor against

bluescreen at a blue desk. LightWave was to provide the background and virtual desk to make this actor appear as if he were a news broadcaster on a high tech set. We wanted to be able to see the lighting as it fell on the blue desk and the actor so we could accurately mimic that lighting in the virtual set. This was the primary reasoning behind the LightWave Laptop approach. The technique also served as a wonderful previsualization tool making the client and the director feel very high tech and confident that the shot would be a success. Again, this is important if you want to be able to charge more for your work in the future. In this particular case it may have been slightly overkill, but we already had the laptop, and the additional hardware was inexpensive enough that it made it worthwhile. The final shot was a success, and this was a contributing factor, so perhaps it was necessary. On a more complex shoot this technique could prove to be a huge time and money saver. If you have ever had to deal with problem footage in postproduction, you may find yourself wishing you had taken this step during principal photography.

On the set, the LightWave scene was modified to match the live action set up. Camera and lighting positions were easily adjusted. In this case, the camera was locked off and the actor was stationary, which greatly simplified our job. We simply ran a composite video cable out of one of the monitors on the set, and fed it right into the CapSure card. This allowed us to grab QuickTime movies (useless to

Continued on page 47

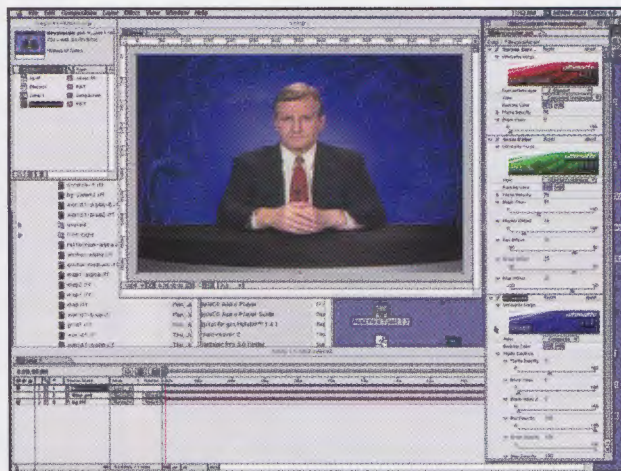


Figure 3: Compositing in After Effects.

ON TOPIC

by David Warner

SIGGRAPH '99 is in the history books now, and if you are a LightWave user it certainly was a show to remember. In fact, I would dare say that this SIGGRAPH was only memorable if you were a LightWave user. As I walked the show floor, looking for 'the next cool thing,' I really didn't see anything to get excited about. Until, of course, I got around NewTek's booth and all the cool, new, and exciting things that were unveiled in that area!

By now you've probably heard a lot about LightWave version 6.0. Personally I was very pleased with everything that was shown in the new version. Having been re-written from the ground up, version 6 addresses many shortcomings that plague the current version of LightWave, and a large number of potentially revolutionary new tools have been added. The new version promises to make life much easier for LightWave animators and I eagerly await its release. However, this is one case where it would be prudent for NewTek to wait until the software is truly ready before releasing it. An announced release date of "third quarter 1999" suggests that we will have version 6 before the end of September, but my gut feeling is that this will not happen; if it does happen, then we are going to be eagerly awaiting a series of bug fixes later this year. I hope I'm wrong.

Our friends at Dynamic Realities were showing off some great new LightWave plug-ins such as NatureFX and Impact 2. As usual, this

was a difficult booth to get close to because they make so many cool plug-ins, but I really liked what I saw from NatureFX. This plug-in really sets a new standard for water renderings in LightWave. Rumor has it that it is really good at doing volumetric clouds too!

Station X Studios has set the LightWave mailing list on fire with its new plug-in, Project: messiah. With an impressive list of features and an incredibly fast and intuitive interface, this plug-in promises to make character animators' lives much easier. This is a custom tool that was written specifically for use by top-notch animators. We should consider ourselves very lucky that the folks at The Station have decided to make it available to the rest of the LightWave community. I was lucky enough to get a semi-private demo of this plug-in's capabilities. All I can say is, if you do character animation with LightWave, then Project: messiah should be at the very top of your 'to buy' list of plug-ins!

There were a few other LightWave developers showing off some new tools, but I didn't have a chance to really check them out. Of particular

interest was a motion tracking plug-in for Aura and LightWave, but information on this product was scarce and there wasn't a running demo of it anywhere.

Unlike shows in the past, NewTek did not have a third-party developer area, so many of the smaller developers just didn't have a presence at this show. In place of the

developer pods, NewTek was pushing Aura 2.0 and the VideoToasterNT, which both look very good, but there's no doubt that this show was really dominated by LightWave. And for the record, I really like the new LightWave [6] logo!

Now here is this issue's selection of questions and answers from the LightWave mailing list.

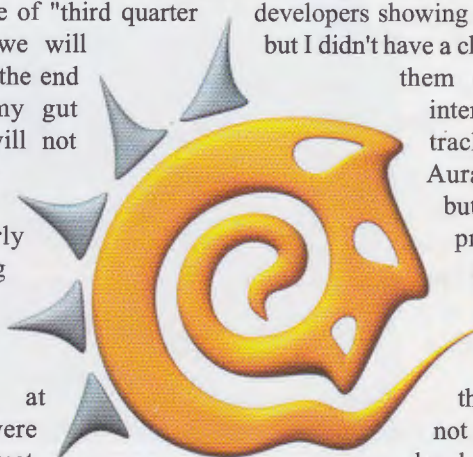
Q Why does the wireframe preview player in Layout not seem to work very reliably? I can never get a preview to play smoothly or at the correct frame rate I have specified. When I render out the animation, it always seems to play faster than the preview shown in Layout.

A Ever since LightWave moved beyond the Amiga platform, it has had problems playing preview animations at the correct speed. You can get a more usable playback if you resize the Layout interface down the smallest possible size, but you will almost never get an accurate frame rate. The best way to get an accurate preview is to go to the Render panel, select the Wireframe or Quick-Render mode and then render out an AVI video clip.

Q How can I add frames to the beginning of my animation without messing up my initial keyframes?

A The Shift Keys function will allow you offset any or all of the keyframes in an animation, except for the one at frame 0...this means that you need to set a "holder" keyframe for the data held at frame 0 in order for a complete frame shift to work correctly.

If you are shifting keyframes for one object, then you can just go to frame 0, create a new keyframe at frame 1, and then turn on the Linear Spline Control option to hold the object in place from frame 0 to frame 1. Open the Motion Graph and then use the Shift Keys function to offset the frames in your animation. This function will ignore the data at keyframe 0 but will shift keyframe 1. With the Linear spline control option turned on, the object will remain stationary in between these two keyframes.



If you are working with many objects, it may not be practical for you to apply the above steps. It is easy enough to create a keyframe for every object in the scene, but it could be a real pain to go through each object and turn on the Linear Spline Control option. In this case, another approach may be easier. First, offset all the frames in your scene by using the Shift Keys function in the Scene Editor panel. This will offset keyframes for every object in your scene. Once you have shifted your animation by the desired amount, go back to frame 0, press Enter to bring up the keyframe creation tool, and highlight the All Items option to create a keyframe for everything in your scene. The new keyframe number should be the same as the number by which you shifted all your frames. Create another keyframe for All Items, but this time enter a keyframe number that is one digit less than the previous keyframe number. Your results should be two keyframes with the same position data, one frame apart from each other. LightWave interprets this as pretty much the same as turning on the Linear Spline Control option, making all of the objects in your scene stay in place from frame 0 to the shifted beginning of your animation.



Figure 2: The Shift All Keys tool allows you to easily move keyframe data for all objects in your scene.

While adjusting the Limited Region settings for my scene, I accidentally messed up the region limits and defined an area that is too small to change. How can I reset the Limited Region settings so that my scene renders correctly again?

Unfortunately, there is no Reset button for the Limited Region Render setting. The only way to reset these values is to open the scene file in a text editor and delete the lines that define the Limited Region settings. Look for two lines that resemble the following:

```
LimitedRegion 1
RegionLimits 0.642000 0.652000
0.562000 0.572000
```

LightWave will not miss these two lines,

so you can delete them and still have the scene file load correctly. If you define new Limited Region settings for your rendered image, those new settings will be added to the scene file the next time you save it.

Looking in the Render panel I see an option to set the number of threads that LightWave uses. My computer has a dual-CPU setup and everything works fine when I have this option set to 1 thread, but as soon as I bump it up to 2 or 4 threads, I see that a number of render errors occur and my render times only decrease by a very small amount. What's wrong here?

First of all, I should shatter the myth that adding a second CPU to your machine will automatically double your computer's render speed. In some cases you will see a very significant decrease in render times, but usually the speed increase falls below 50%, sometimes even below 30%. That second CPU will come in handy when you're doing a final render and you set up a ScreamerNet node for each CPU, but this can require a lot of memory, and due to system overhead you still won't see your scene render twice as fast.

As for the threads question, you should never increase that value to more than two threads because you only have two CPUs. If you had a (really expensive) workstation with more than two CPUs, then the higher thread settings would come in handy. However, since your computer does not fall into this category, you should avoid these higher settings. Even with Threads set to 2, you will still find that many plug-ins and post

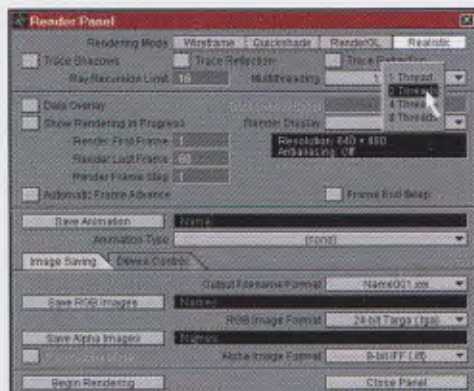


Figure 3: The Multi-Thread rendering option is accessed from the Render panel.

processing effects do not always work the way they are supposed to. This problem should be corrected in LightWave 6, but the current version definitely suffers from some multi-thread rendering problems.

I have an object following a motion path with the Align To Path function turned on. The object moves the way I want it to, and the Align To Path function works perfectly until it reaches the last frame in the object's motion path. At that point, the object's rotation snaps back to 0 on all axes. How can I keep the object aligned to its path even when there isn't a path left for it to travel along?

The Align To Path function works by looking ahead in time to see where the object will be located in the future, and rotates the object to be aligned to that future position. The default setting is to look ahead one frame, but this value can be changed. Your problem is that when the object reaches the end of its motion path, the Align To Path function has no new position to align the object to, so the object just snaps to its last keyframed rotational value.

The last part of that sentence holds the solution to this problem. All you have to do is create a new keyframe for your object at the end of your animation. If you jump back one frame from the end of the animation, then make a note of the rotational values created by the Align To Path function, you can manually apply these rotational values to the object in the last frame. Now when the object reaches the end of its motion path and the Align To Path function runs out of data to apply to the

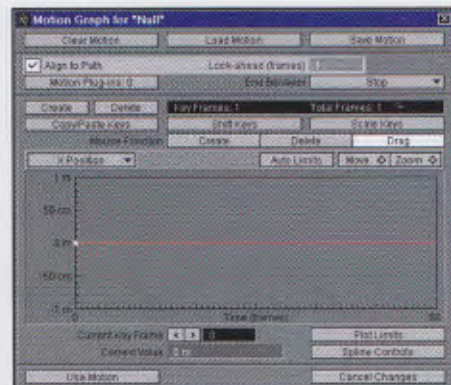


Figure 4: The Align To Path function is accessed from the Motion Graph.

object's rotation, Layout will apply the keyframed rotation values and the object will stay aligned the way you want it to.

Q I can't get the QuickTime saver to work. Sometimes when I select "QT Saver" as my animation type, LightWave crashes. Other times I get an "Initialization Failure" message when I start to render my animation. How can I save my LightWave animations in the QuickTime format if the QT saver won't work?

A It is not a good idea to render your animations directly to an animation file, because if something goes wrong during the render, you will end up with a video clip that is unplayable and will have to re-render the entire animation. It is much wiser to render your animation as individual images (also known as an 'image sequence')

and then assemble these images into a video clip.

The QuickTime saver in LightWave has long been a source of frustration for many people. Nobody has ever really nailed down why it works for some people and not for others. After many months of struggling to get this thing working on a number of different machines, I finally gave up and decided to find a better way.

As far as I'm concerned, the best tool to turn your rendered image sequence into a QuickTime video clip is Apple's QuickTime MoviePlayer. You will need to pay about \$30 to unlock the save/export functions of the QT MoviePlayer, but it is money well spent because you will have access to some very powerful compression codecs. Upgrading your MoviePlayer program is

very easy, and can be done online in a matter of minutes. Once you have the full version of MoviePlayer running, all you have to do is import the rendered animation image sequence, then export it as a QuickTime video clip with the compression codec that you prefer. (I highly recommend the Sorenson codec for everything.)

That'll do it for this installment of On-Topic. If all goes according to NewTek's plan, I hope we will be tackling our first batch of LightWave v6 questions in the next issue!

For more info about David Warner or TutorVision 3-D CD, his series of multimedia tutorial CD-ROMs for LightWave, please visit <http://www.tv3d.com> or send some e-mail directly to Dave at dwarner@tv3d.com.

The BEST just got BETTER!

Now with 4 different volumes to match your skill level as you learn how to use LightWave and Inspire 3D...

TUTOR VISION
3-D CD

- 1** The FIRST video tutorial CD ever released for LightWave! This disc contains nearly 4 HOURS of video tutorials that focus on doing many science fiction effects as well as more practical every-day effects. Includes a video interview with Alan Chan, dozens of images from other LightWave users, a dozen objects, a dozen texture maps, and MUCH more!
- 2** The most comprehensive LightWave CHARACTER ANIMATION tutorial CD ever created! This 2-disc set has more than 6 HOURS of video tutorials and will teach you everything you need to know to start doing Character Animation with LightWave 3D version 5.5 or greater! Includes an interview with Stephen Bailey and MUCH more!
- 3** More than 5 HOURS of video, with many new techniques for LightWave and Aura! Topics covered include advanced motion control tricks, beginner and intermediate compositing techniques, using SkyTracer and Steamer effectively, tapping into the power of HyperVoxels & ParticleStorm, and more! Also contains product reviews in video format!
- 4** The ULTIMATE set of introductory tutorials for new LightWave and Inspire users! This CD will demonstrate many basic concepts and techniques that new users will need to know before they can get the most out of these incredibly powerful animation tools! The most Frequently Asked Questions are answered in easy to understand video and text tutorials!

Only \$49.95 each! Call or check our web-site for multi-CD discounts!

Recommended System Configurations

PC: Pentium, 16mb RAM, audio card,
4x CD-Rom, Win95/WinNT

DEC Alpha: FX!32, 16mb RAM, audio card,
4x CD-Rom, WinNT

PowerMac: 16mb RAM, 4x CD-Rom,
System 7.5 or greater

*Visit us on the
World Wide Web
for FREE Demos,
Program Updates,
and LOTS of
other Cool Stuff!*

Event Horizon Studios

Phone: 518-756-6850

FAX: 518-756-6836

E-Mail: info@tv3d.com

Web-Site: <http://www.tv3d.com>

We Accept:    



Dynamic Realities drew crowds with NatureFX.



Don Ballance demoed Aura and the new ToasterNT.



NewTek's demo stage.



Keyframes went fast!



NewTek had record crowds for L[6] demos.



Messiah screen shot.



SI99GRAPH
Los Angeles

photos courtesy of Jared Bents



Doug Kelly and Ace Miles.



Doug Kelly getting a messiah demo.



On A Wing and A Prayer

Thay-Lore Character Setup

by Graham McKenna

Thay-Lore is a character I created for a personal project that will demand photorealistic full body and facial animation. The success of this project relies on the correct character setup procedures.

My first step was to examine the features that were needed to allow me to animate the character convincingly, particularly the wings. Thay-Lore also has the usual human features, namely two legs, two arms, fingers, etc. I had to construct the mesh to conform to the Puppet Master routine that I would perform later to create the skinned mesh. Figure 2 shows the wireframe view of the resulting low-density mesh, totaling 5766 polygons.

The face at this point needed particular attention. I had to be sure it was going to animate convincingly before going through the whole setup procedure, or I might end up backtracking just to adjust the face. To compromise between the definition I wanted in the facial features and keeping it simple enough to be animatable, I eventually ended up with 458 points around the front of the head, Figure 3. This at first seemed excessive, but not all of these points need to be adjusted with every morph target. Figure 4 shows two facial poses.

The character's main setup features that differ from normal human setup are the wings. My first worry with this was how to structure the model to prevent gimbal lock. I made a few tests with boxes laid out in different ways in Modeler. This let me try the hierarchy of the wings in layout until I found a solution.

This approach helped speed up the setup process by

isolating a known problem, find a solution, and then apply that solution to the character. Figure 5 shows the wings in their final rest position. This layout allows each wing section to use all three axes of rotation. The OpenGL window shows the small purple wedge-shaped section that helps create a sharp fold or bend halfway down the length of the wing. This is especially useful for the attack poses so typical of this character. The rest of the sections are evenly spaced along the wing to allow a smooth flapping motion when the character is in flight.

The character's hands went through the same testing process to give each joint freedom in all three axes of rotation. This resulted in each finger's rest position pointing along the -Z axis. Each finger has three sections, the base, middle and tip. I tested every other part of the figure in a similar fashion, then applied the solutions to the main mesh. Figure 6 shows the final low-res model. The colored polygons will become unique objects. The color-coding has no significance other than being a visual cue to where the mesh is split when setting up Puppet Master in Layout.

I saved this model as the anchor mesh that Puppet Master references when skinning a blend over the hierarchy of sections. I then

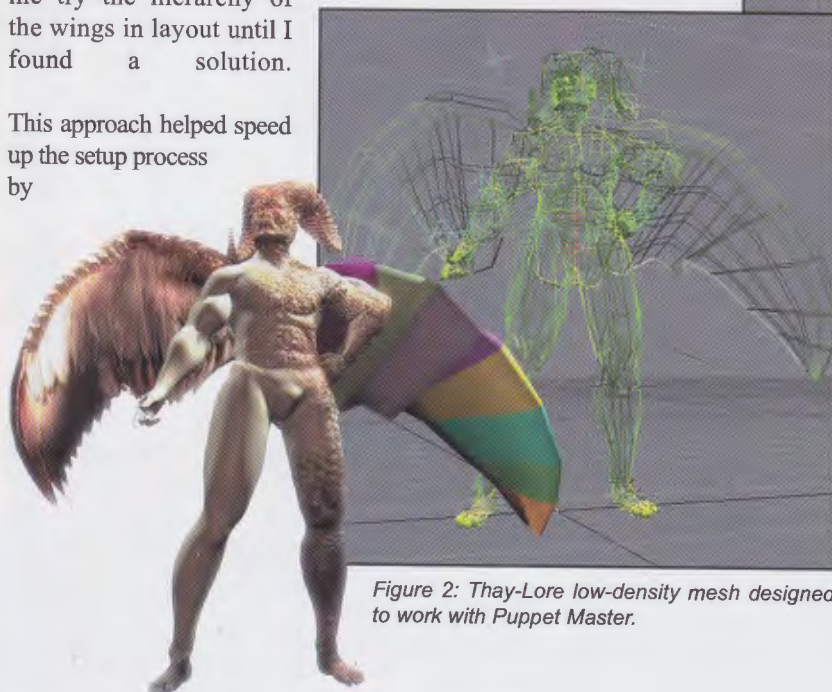


Figure 2: Thay-Lore low-density mesh designed to work with Puppet Master.

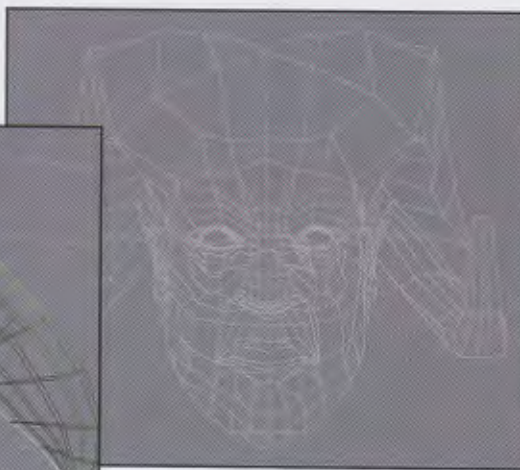


Figure 3: Optimized face mesh of 458 points.



Figure 4: Two facial expressions for Thay-Lore.

Figure 1: Thay-Lore character showing range of surfaces from OpenGL color coding to final colors and textures.

selected these colored polygons by surface and cut them to a new layer, where they were saved out as unique objects in the same directory as the anchor mesh. The next step was to run the Puppet Master batch routine. This creates a .sec file for every unique object found within the directory.

My next step was to load each object, starting at the groin and working out, so the object list is in a logical order. I also created 16 nulls that are included in the hierarchy as control points. With all objects loaded, I started placing the pivot points and creating the hierarchy. Once hierarchy and pivots were in place, I applied the Puppet Master plug-in to the objects. Each section object needs to have the Puppet Section plug-in applied to it, and the anchor object needs the Puppet Master Anchor part of the plug-in applied. Back in Layout, the anchor object displaces points to conform to the referenced section object. For example, when the calf section is rotated on the Pitch axis, the Puppet Master plug-in knows to skin a blend across the knee where the calf's points split off. At this point I played around with the areas around the joints until the skinning looked correct. Pivot point placement is crucial at this stage; refining the pivot for each joint helps achieve a better crease. Due to the extensive earlier testing, this process soon produces the final skinned anchor.

Once the anchor was finalized, I made all the sections invisible by checking the Unseen by Rays and Unseen by Camera options and turning off all shadow options in the Objects panel. In the Scene Editor, I hid all objects except the anchor mesh and the 16 control points, which leaves a minimum of objects cluttering up the layout window when animating.

With Thay-Lore now successfully skinned, I applied a mesh smoothing routine to eliminate any faceted curves when the character is deformed into extreme poses. The first step to this procedure is to go back to Modeler to create a higher resolution character from the anchor object. The new target object is created using the Metaform Plus plug-in with the settings shown in Figure 7. The first two check boxes are ticked and the third left blank; the first option is already tripling all quads, which will fix all non-planar polygons. The subdivision level controls the detail of the new mesh. Thay-Lore is set to one subdivision, as anything higher would exceed LightWave's 65,000-point restriction.

The resulting mesh has 42144 polygons; the original model has 5766 polygons. This massive difference in polygon count is the main advantage of the whole Metaform procedure: it allows the use of the anchor object for animating purposes, then smooths the mesh at render time to the higher level. Having such a low polygon count can result in real-

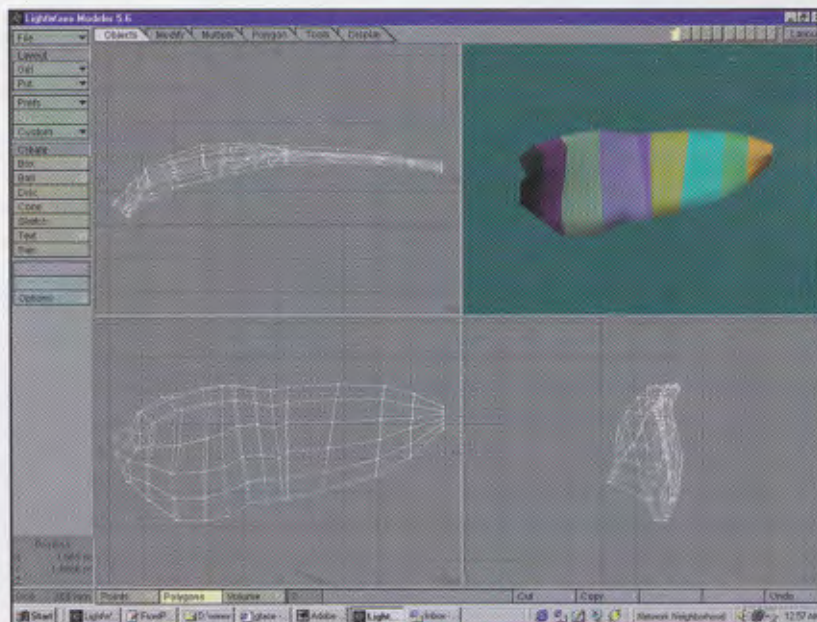


Figure 5: Wing structure oriented to prevent gimbal lock and provide a built-in center fold.



Figure 6: Thay-Lore mesh color-coded for Puppet Master sectioning.

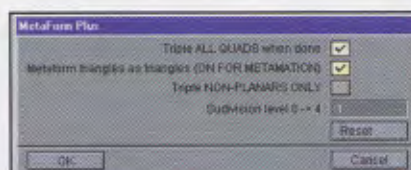


Figure 7: MetaFormPlus settings for Thay-Lore anchor subdivision.

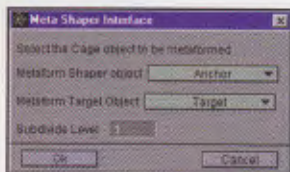


Figure 8: Metamotion Shaper (cage) and Target settings.



Figure 9: Thay-Lore in smoothed OpenGL mode.



Figure 10: Leg modeled to minimize IK flailing.

time previewing (with the right hardware) when used in conjunction with the bounding box point/polygon threshold settings in the Options menu.

My next step was to load the new target in Layout and apply the Metamotion plug-in to the anchor and new target. Figure 8 shows the dialog boxes for both anchor and target. Make sure the subdivision level

in the anchors dialog box is set to the same level that was chosen in Modeler, if you don't want your scene crashing.

Figure 9 shows the finished smoothed mesh. This is the next opportunity to refine the look of each joint by tweaking the placement of pivot points.

With the Puppet Master and Metamotion plug-ins set up, the character is nearly ready to animate. The next stage is the setting up of control points and inverse kinematics (IK). Figure 2 shows several nulls around the character. To enable these nulls to function as controls, I chose to apply a motion plug-in, Relativity. This plug-in is used to tell each section object to match the rotation of the section's goal. This allows all the sections within the puppet to be hidden from the Layout view, leaving only the anchor and control points visible. This allows a 6000 point/polygon threshold, which on most systems will enable real-time screen redraws while posing the character. This makes animating the character much easier and faster. I also set the control points to be color coordinated, with a certain color for one side of the figure and a different color for the other side. This provides an easy visual reference when animating.

I began the IK setup by creating two new nulls, naming them for right and left, and color matching them with the other controls. I placed them at the ankle of each leg. I then chose where to break the IK chains' influence, in this case just above the groin, by accessing the IK panel and ticking the Unaffected by IK of Descendants check box. This means that anything above these sections in the hierarchy will not be affected by the IK null's behavior. I

then applied IK to the sections I did want to effect, forming an IK chain down each leg. I chose each ankle and set them to Full-Time IK and Match Goal Orientation. For this character (and in general, most characters) there is no need for limit settings. By taking IK into consideration way back in the modeling stages, you can build a responsive, accurate IK chain. If you bend the character's legs in Modeler, you can make calculations easier for the IK to solve by giving it a priority over what is negative and what is positive. If the legs are modeled straight, the IK solver seems to treat this as 'in the middle' and often results in flailing and general posing difficulties. Figure 10 shows all that is needed to allow easier calculations.

The result of these procedures is a character that will withstand demanding poses necessary for many animations. You can see an example of this character in motion in the ThayLore.MPG video clip on the Keyframe CD-ROM. In Figure 11, Thay-Lore looks to the light for inspiration for what new life we are able to breathe into him.

Graham McKenna can be reached via email at graham@grahamprismatic.free-online.co.uk, or you can visit his website, www.grahamprismatic.free-online.co.uk.



Figure 11: Thay-Lore character set up, animated, lit and rendered.

This image was from an animation for the 1 Million AD Trailer for Ruddy/Morgan Productions at Threshold Digital Research Labs (T.D.R.L.). The trailer was about 28 shots and involved creating and animating a Triceraphant (half elephant, half triceratops) and a Batadactyl (half bat, half pterodactyl). Mike Hatfield and Matt Morgan took on the Triceraphant and I battled the Batadactyl. This shot was part of an animation of the Batadactyl getting startled from his afternoon nap by the stampeding Triceraphant. The Batadactyl shakes around a bit and opens his wings for flight.

To model the Batadactyl, I used MetaNURBS and point-by-point modeling. I modeled the body for Puppet Master with his wings out flat and his feet pointing down to make texturing easier. Texturing was pretty straightforward. I took a snapshot in Modeler for his body and wings, and one for the side of his face. I painted textures for the front, back and side of his body and the top and side of its head and blended them using texture alphas. I had Layout open on one monitor and Photoshop on the other and went back and forth comparing the two screens for various texture adjustments. I had the base textures done in a couple days and was ready to take off for the weekend. The next week I was back in Modeler. I saved out the Puppet Master sections and set up the Bod file. Using Fiber Factory, I made some whiskers and hairs on parts of its body and saved out different patches to parent to the sections. I then made more surface and texture adjustments and made the wings translucent to get the light glow through them. I made some model tweaks to the Puppet Master sections and morphed them for the changes to take place.



For this shot I grabbed a tree limb made in Tree Druid and used a leaf texturing technique Matt came up with (see the Leaf It To Me! tutorial on the next page) to map the individual leaves. I painted up a blurry background image in Photoshop and mapped it on a polygon. I set up a really bright sun light to give it a warm rim and gave the light a soft glowy flare to shine through the trees and cover up part of the creature. I also placed a blue-green fill light underneath the creature to lighten up the dark areas a little bit. What really made the image stand out was adding glow to the background polygon. This wrapped the light around the objects and softened up the picture. Then I added film grain and that was that. It was a really fun project to work on. Of course, any project is fun when you get to run around the office and act like a bat for inspiration!

Jeremy Yates is a digital artist at Threshold Digital Research Labs. You can send email to Jeremy at jeremy@thethreshold.net.

Morphing and Texture Mapping Leaves



Leaf it to Me!

by Matt Morgan

Have you been frustrated while trying to make photorealistic leaves in TreeDruid? Since TreeDruid was released, many a user has hunted for a solution to this problem. The software offers a few different leaf choices, the better ones being expensive on the point count. Why can't users simply make alpha, color, and bump maps? Well, you can! You can create awesome texture-mapped leaves using morph targets and an alpha map.

First, make a tree in TreeDruid (www.dynamic-realities.com) or any other procedural tree modeler (Like ONYX' Tree Pro, www.OnyxTree.com). Use the rectangle or plane option for the leaves, as shown in Figure 1.

Load the tree into Modeler, and cut and paste the leaves into a layer of their own. Surface the leaves "Default" and take note of the polygon count, as shown in Figure 2. You need to know the number of leaves you have. Since TreeDruid makes triangulated leaves (two triangles per rectangular leaf), if you divide the number of polygons by 2, the result should equal the number of leaves. Save the leaves as Leaves_m01.lwo. I usually use m00, m01, m02 for morph targets so m01= 100% morph, and m02= 200% morph.

In an empty layer, use the box tool on the Z-axis to make a flat plane in the shape of a perfect square. Make sure your plane is centered at 0,0,0. Surface your plane as "Leaves". Refer to the leaf count you calculated previously; this is how many cloned planes you need. Array/Clone your plane by this number of leaves, minus one for the original plane you already created.

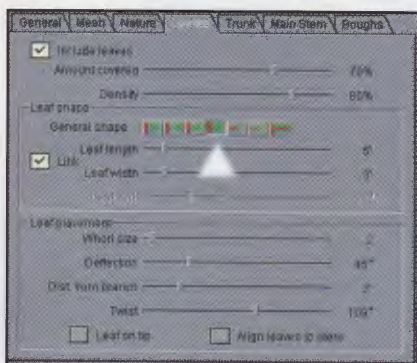


Figure 1: Rectangle leaf setting in TreeDruid.

If you check the point counts, your planes layer should now have the same number of points as your leaves layer, as shown in Figure 3. Save your cloned planes out as Leaves_m00.lwo. Tripling your planes shouldn't be necessary. In my experience, TreeDruid leaves are usually less than 0.01% non-planar. LightWave usually only has problems with surfaces that are more than 0.5% non-planar.

In Layout, load in your trunk and both leaves objects. Parent the leaves to the trunk. Set Leaves_m00 to morph to Leaves_m01 at 100%. Lastly, set object Leaves_m01 to 100% Dissolve, making it invisible when rendered. This morph target setup enables you to apply your leaf maps accurately to Leaves_m00, which

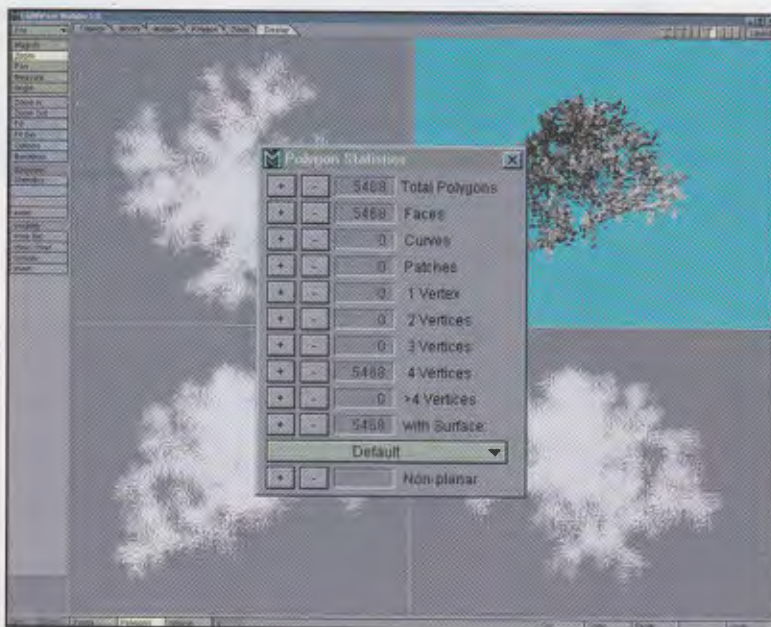


Figure 2: Leaves_m01.lwo. This is the morph target of the leaves for use with the planes.

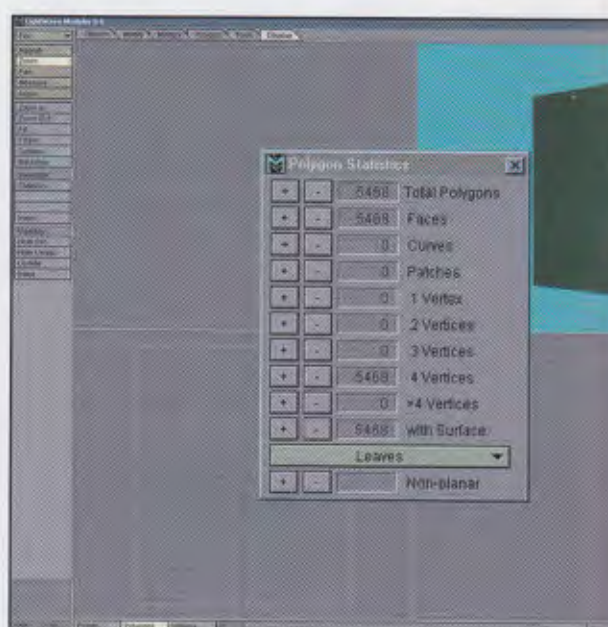


Figure 3: Leaves_m00.lwo. This is the plane for z-axis planar mapping of leaf textures.

has a perfectly flat surface with faces perpendicular to the Z-axis. The Leaves_m01 morph target created by the tree plug-in has faces pointing every which way, and would be nearly impossible to map accurately. LightWave's morphing process keeps the maps aligned to the surface throughout the morph, so the maps stay put.

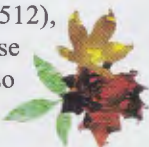


Figure 4: A quick close-up rendering with leaf textures applied.

Now all you need to do is get some textures and alpha maps of leaves and apply them to your leaves. Texture your planes on the Z-axis, and make sure Double Sided is checked. You can use the alpha map for the transparency channel. However, if you want to use Shadow Maps you need to map the leaf's alpha as a clip map, since shadow maps don't pay attention to transparency. Although a clip map has a harsh edge, anti-aliasing should take care of that, Figure 4. You can only get so close to a single polygon leaf before it obviously won't hold up. That's when you 'model' a leaf.

Why does this work? Due to LightWave's process for modeling a plane, your morph target cannot mess up as long as you have the right number of points. Anyone who has modeled morph targets has probably run afoul of those horrible criss-cross morph targets. These problems happen because the point order is different between the target and the source. A simple plane is made of four points. LightWave makes points 1,2,3,4 in that order. When you clone that plane in Modeler, it clones points 1,2,3,4 in that same order over and over. Modeler doesn't make part of plane 6 and then part of plane 3678. This consecutive point rule also applies to a procedural modeling program or plug-in like TreeDruid. So, by using the clone tool, you guarantee that all the cloned planes will have the correct point order.

Texture mapping your leaves makes it easy to use extremely complex leaf shapes that you could never afford to render with geometry alone. This approach also makes it very simple for you to change the shape of the leaves, since your alpha channel defines the leaves' outlines. If you are looking for some good leaf textures, check out Onyx' Tree Professional. It is an awesome tree modeling program that is many times costlier (and more robust) than TreeDruid. It also includes a wonderful library of Leaf image maps, at perfect square resolution (512x512), including alpha and bump maps. These maps include both front and back sides, so if you copy/flip/offset your leaves you can render the back sides as well.



Matt Morgan is a senior animator at Threshold Digital Research Labs. You can email him at violator@thethreshold.net.



Animation Training



3D Exchange

3D Animation, Modeling, and Effects Training

New Classes • Bigger Facility • Better Location

www.3dexchange.com • (510) 747-1540



A BEHIND-THE-SCENES LOOK AT THE CREATION OF OUR COVER

Deep Blue Sea

by Flash Film Works

Can you tell where film stops and LightWave begins? Is that water real? Is the shark real? This summer's thriller *Deep Blue Sea* used a heavy dose of LightWave and NatureFX to bring to life an aquatic lab and its smarter-than-human sharks. How did they do it? From the mouths of the animators...

The Cover Shot

The water was created using Arete's NatureFX plug-in from Dynamic Realities. The water had to be hand tracked from an aerial shot taken by second unit. Tracking of the water and the water lighting and creation of the waves were done by John Coats. 3D architecture of the Aquatica station was created by Don Myers. It was necessary to do the shot in two renders and composite together the NatureFX water. We got the best results by rendering using enhanced low resolution, but it didn't contain enough pixel depth for Aquatica. A second render was done at enhanced high and then digitally composited by Dan Novy, where the airplane was added back over top.



Image © 1999 Warner Bros. Courtesy of Flash Film Works.

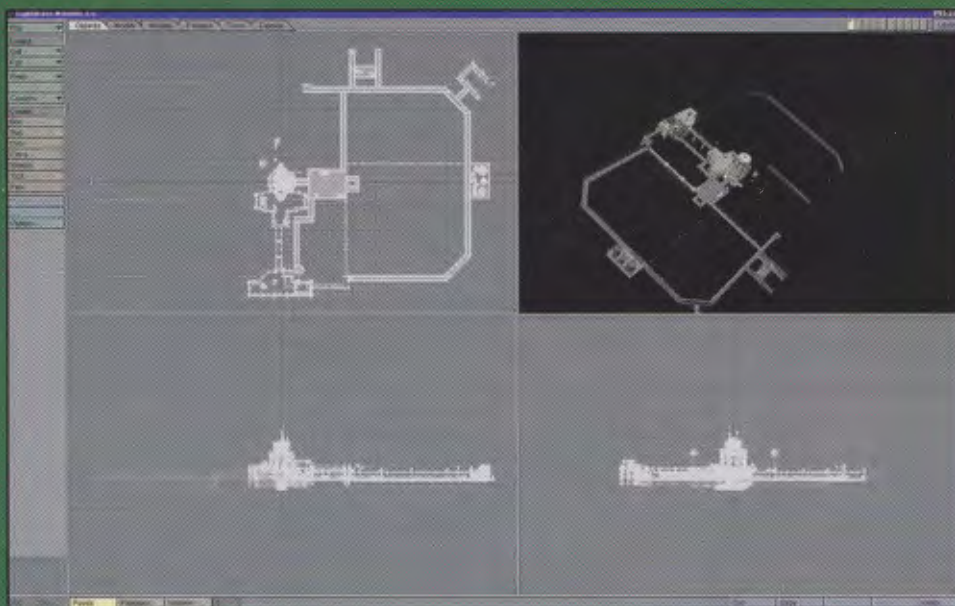


Image © 1999 Warner Bros. Courtesy of Flash Film Works.

The first step in making this shot was to create the 3D geometry of the Aquatica complex, so that the waves could work against it. We built an Aquatica 3D model (over 500,000 polygons), completely mapped from every angle using photographs from the set in Mexico. In this scene, the model was used with the original mapping, and some of the walkways were front projection mapped where shadows and highlights had to match some of the original photos. The 3D architecture was built by Don Meyers. The 3D water geometry as well as the photographic rain maps were built by Dan Novy. Animation of the water, texturing and lighting were done by Bruce Pearson. The helicopter blades, the animation of them breaking and the resulting debris (using Particle Storm) were done by Ken Stranahan.

Right - finished shot. Below - original image.

The water was created using NatureFX. The animation of the water was done by Tim Everitt and Dan Novy. Don Myers created the walkways; they interact with the water. The storm clouds and sky began as Skytracer rendered, and then were matte painted over for the storm by Tim Donahue. White-



Image © 1999 Warner Bros. Courtesy of Flash Film Works.

caps on the water were created from a process created by Dan Novy utilizing the NatureFX swell shader by offsetting two layers of the ocean grid and using one layer to clip the other layer, which gave us a matte for that area of clipping. This matte was then used with real whitecaps and foam and digitally composited into the shot.



Image © 1999 Warner Bros. Courtesy of Flash Film Works.

The water was created using NatureFX. 3D animation of the water, texturing, and white caps were created by Dan Novy, and the Aquatica structure was created by Don Myers. This again required us to do two renders, the 3D water to be rendered at enhanced low and Aquatica at enhanced high. We also had glow passes for brightening the rain elements in compositing as well as moving rain over the lights using Particle Storm. The large waves and all the rain elements were digitally composited by David Fogg and the sky background and lightning were matte painted by Tim Donahue.



Image © 1999 Warner Bros. Courtesy of Flash Film Works.



were created by John Coats. Shark animation was performed by Ken Stranahan and Don Myers. The shot required roto work of the shark entering the compound and was digitally composited by John Mesa.

Image © 1999 Warner Bros. Courtesy of Flash Film Works.

The top image is the original shot, the bottom is the finished piece.

The water and the shark's wake were created using NatureFX. 3D water, animation, and tracking



Image © 1999 Warner Bros. Courtesy of Flash Film Works.



www.flashfilmworks.com

Left to right: Penny Mesa, VP of Flash Film Works, Lincoln Kupchak, Visual Effects Editor, Bill Mesa, President. Cool toys in the office are from some of the films Flash has worked on.



Left: Dan Novy, Technical Supervisor, viewing a movie. Above: John Mesa (sitting), Co-Visual Effects Supervisor and Lincoln Kupchak, Visual Effects Editor viewing a composite of a 3D LightWave render of Aquatica.



Building Image Processing Plug-ins For LightWave, Part II: Put on a GUI Face

by Robert Gougher

In last issue's article, we explored SLIP: the Simple LightWave Image Processor. The purpose of this generic image processing plug-in is twofold: first, it illustrates how to build plug-ins for LightWave's image filter interface. Second, it serves as a basis for building more sophisticated LightWave plug-ins. So far, we have described three different versions of SLIP. First, a pixelwise processor that does image processing operations pixel-by-pixel. Second, a planewise processor that uses temporary imagebuffers to process an entire image at a time. This is necessary for operations such as masks, which require more than one input pixel of information for each output pixel of information. The third version employs a log file to allow you to spew information for tasks such as debugging or profiling. In this issue, we will pull these three versions together into a single plug-in unified by a graphical user interface (GUI). The GUI is built with the calls in the LightWave SDK's `lwpanels.h` header file.

STEP 0: Plan your work.

In order to combine all this functionality into one plug-in, and then expand upon it a bit, we must first do some planning. What we want is a GUI which enables the user to apply all of the pixel operations and all of the plane operations, as well as an option to output logging information. We would also like the plug-in to be extensible, so it can be changed and improved in the future.

Figure 1 is one embodiment of these requirements. There are three main sections to this GUI: The basic controls, the algorithm controls, and the settings controls.

The basic controls include saving and loading the information for the plug-in alone. Note that these operations are distinct from the load and save operations that you define for LightWave to use when saving a .lws scene file. However, inspection of the relevant routines (`SLIPsave`, `SLIPload`, `cback_butLoad`, `cback_butSave`) will show that much code is shared between them. Also included in the basic controls is a click

box for activating the log file, and the ever popular About box. If you are going to write plug-ins for fun and/or profit, you definitely want an About box to let people know who you are.

The second section of the GUI contains the algorithm controls, which include

pulldown menus for the algorithm type (pixel and plane), as well as pulldowns for the algorithms themselves. Also included are click boxes for making the output a negative image and activating bounds. As we saw in last issue's article, bounds checking is necessary so that RGB values above 255 are set to 255 and values below 0 are set to 0. If we don't do this, out of bounds values may "wrap around" creating dark values where light should be and vice-versa.

Finally, the settings controls are those particular to specific algorithms. This includes controls for specifying a value to add, a factor to apply, or mask values to apply in a planewise fashion. Additional controls could be added to this set as the set of pixel and plane algorithms grows. You may also notice that in addition to the three control sets described above, the GUI in Figure 1 contains a folder tab control with tabs for the algorithm controls and the settings controls. Once we get all of the individual controls working, we will use these folder tabs to switch between algorithm and settings controls, thus making more efficient use of screen space.

At this point, you probably want to print out a copy of the source for `slipGui.c`, or even better, load it up in an editor that has a search function. Almost all of the code discussed in last issue's article has been modified or expanded to accommodate functions we are adding in this article. Many of these changes are not worth calling out explicitly, but are better left as an exercise for the reader, if you really want to get down to the nitty gritty. Another reason to have a printed or electronic version of the code handy as you read is that I'll be

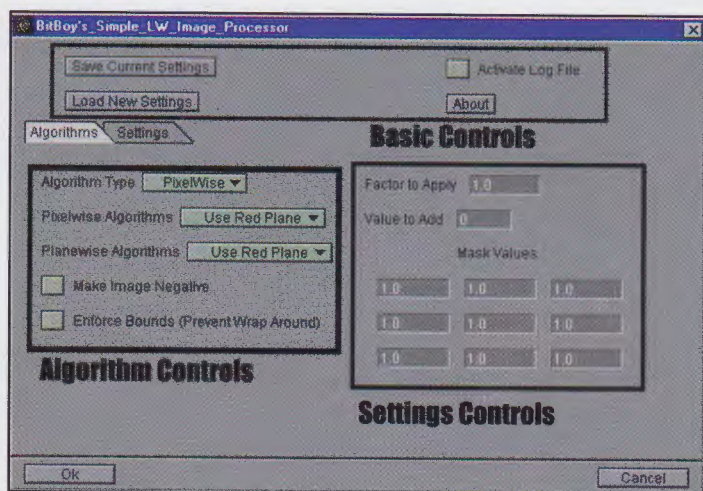


Figure 1: One possible GUI for a unified SLIP plug-in.

referring to quite a few of SLIP's routines without actually including them in the article. A final good reason to get the code is to experiment with some of the calls; maybe you will come up with a more efficient or better-designed GUI by making some of your own modifications. The code is on the Keyframe CD-ROM for this issue, but you can also download it from my website. Just go to www.robougher.com/keyframe.

Step 1: Expand SLIP_DATA and its initializing function, SLIPcreate.

The first thing we need to do is expand our SLIP_DATA structure, which supplies the controls for our GUI as well as the data members corresponding to those controls. We also need to expand the SLIPcreate function so that all of those members are initialized. Take a look at the SLIP_DATA structure, as well as the SLIPcreate routine to see how these are defined and initialized. Note that there are controls and data members corresponding to each piece of the GUI. Note also that almost all data members are integers, except for those corresponding to controls which hold floating point values (e.g., factor or maskValues) and the strings holding filenames. Notice also that the defaults for the pulldown menus and click boxes are set to the first entry (entry 0) and that the file operations assume that files will be on the C: drive and have the .slp extension.

Step 2: define CSHUR routines.

The next thing we need to do is define some create, set, hide, unhide and read (collectively referred to as CSHUR) calls for each of our three control sets. Modularizing the atomic calls in this way will make it easier to understand what the plug-in's main interface routine is doing.

These calls do pretty much what you would assume from their names. Most of these routines are quite simple. The set routines (setBasicCtrls, setSettingsCtrls and setAlgCtrls), set the value in the control according to the value in the SLIP_DATA structure. The read routines (readBasicCtrls, readSettingsCtrls and readAlgCtrls) do the reverse of this. They copy the value in the control, which may have been changed by the user, to the corresponding data member within SLIP_DATA. The hide routines (hideBasicCtrls, hideSettingsCtrls and hideAlgCtrls) erase the controls, thus making them invisible and unusable without actually destroying them. The unhide routines (unHideBasicCtrls, unHideSettingsCtrls and unHideAlgCtrls) re-render and redraw the controls in question, making them visible and usable once again.

The create routines (createBasicCtrls, createSettingsCtrls and createAlgCtrls) are a bit more complicated. They initialize each control, set up any callback functions (more on these later) and move the control to the appropriate place on the GUI. You'll notice that the pulldown

menu is supplied with a global array of strings during initialization. This global array is the list of choices for the pulldown. Also defined is a global array of enums which correspond to the strings in question. This enum array can be used later to check which popup option was selected. Code Snippets 1-4 illustrate this technique.

Code Snippet 1: The global array containing the strings the user sees on the planewise algorithms pulldown menu.

```
static char *planeAlgs[] = {
    "Use Red Plane",
    "Use Green Plane",
    "Use Blue Plane",
    "Apply Factor",
    "Add Value",
    "Apply Mask",
    NULL;
}
```

Code Snippet 2: This is the global enum used when setting or reading the planewise algorithms pulldown menu from within the code.

```
typedef enum {
    PLANE_RED_PLANE,
    PLANE_GREEN_PLANE,
    PLANE_BLUE_PLANE,
    PLANE_APPLY_FACTOR,
    PLANE_ADD_VALUE,
    PLANE_MASK,
    PLANE_NULL
} t_planeAlgs;
```

Code Snippet 3: An actual control creation command from within createAlgCtrls. The arguments are, from left to right, the panel functions pointer, the panel ID, the string to be placed next to the control, and the array of strings to populate the pulldown menu itself.

```
SLIP_DATA->ctrl_popPlaneAlgs = POPUP_CTL(SLIP_DATA->panel, SLIP_DATA->panelID, "Planewise Algorithms", planeAlgs);
```

Code Snippet 4: A switch statement employing the enum. Note that the default should never be reached.

```
switch (defInst->planeAlg) {
    case PLANE_RED_PLANE:
    case PLANE_GREEN_PLANE:
    case PLANE_BLUE_PLANE:
    case PLANE_MASK:
    case PLANE_APPLY_FACTOR:
    case PLANE_ADD_VALUE:
    default: // THIS SHOULD NEVER HAPPEN!
} //end switch block
```

Once our CSHUR routines are defined, we must modify SLIPInterface. These modifications are simplified by the use of the CSHUR routines. Instead of calling every control explicitly, in many cases we can simply call the CSHUR routines to service all controls in a group. Careful inspection of the code for SLIP will reveal that you can't always depend on the CSHUR routines, but they suffice most of the time. The interface function uses the CSHUR functions in the manner indicated in Code Snippet 5.

Code Snippet 5: After setting the size of the GUI, the CSHUR functions are called.

```
PAN_SETH(defInst->panl, defInst->panID, 390);
PAN_SETW(defInst->panl, defInst->panID, 600);

createTabs(defInst);
createBasicCtrls(defInst);
setBasicCtrls(defInst);
createAlgCtrls(defInst);
setAlgCtrls(defInst);
createSettingsCtrls(defInst);
setSettingsCtrls(defInst);

hideSettingsCtrls(defInst);
ERASE_CON(defInst->ctrl_popPlaneAlgs);
```

The new code uses the create, set, read and hide functions to set up the initial state of the UI. Note that it also uses the ERASE command to explicitly hide the plane algorithms popup. This is because the initial state of the algorithm type popup is set to pixel algorithm, and so we want only the pixel algorithm popup to be shown. As we will see in step 4, other routines will use the hide and unhide controls to change the visibility of certain controls based on the settings of other controls. However, we must first link our data collected from the GUI with the SLIPprocess function, which actually modifies the images produced by LightWave's renderer.

STEP 3: modify the process function to use the new controls.

By defining these CSHUR routines, and calling them in the SLIPInterface function as in Code Snippet 5, we get to a point where executing the SLIP plug-in will produce the GUI shown in Figure 1. Now that we have a displayable GUI that collects information from the user, we must collect that information and use it to drive the algorithms. Within SLIPprocess, a collection of conditionals and case statements surrounding various image-processing operations accomplish this task. Code Snippet 6 illustrates the general design of the SLIPprocess routine.

Code Snippet 6: The general structure of the SLIPprocess routine.

```
if (defInst->logFile) {
    OPEN AND INITILIZE LOGFILE
}

if (defInst->algType == ALGTYPES_PIXEL) {
    for (i = 0; i < fa->height; i++) {
        for (j = 0; j < fa->width; j++){
            switch (defInst->pixelAlg) {
                case PIXEL_RED_PLANE:
                case PIXEL_GREEN_PLANE:
                case PIXEL_BLUE_PLANE:
                case PIXEL_APPLY_FACTOR:
                case PIXEL_ADD_VALUE:
                default: //ERROR: SHOULD NOT
HAPPEN!
            }
        }
    }
} else if (defInst->algType == ALGTYPES_PLANE) {
    inImageBuffer = newImageBuffer(fa->height, fa->
width, 1);
```

```
    outImageBuffer = newImageBuffer(fa->height, fa->
width, 1);

    //Loop through the lines in the image to load
framebuffer

    switch (defInst->planeAlg) {
        case PLANE_RED_PLANE:
        case PLANE_GREEN_PLANE:
        case PLANE_BLUE_PLANE:
        case PLANE_MASK:
        case: PLANE_APPLY_FACTOR:
        case: PLANE_ADD_VALUE:
        default: // ERROR: SHOULD NOT HAPPEN!
    } //end switch block

    //Loop through the lines in the image to write
framebuffer

    freeImageBuffer(inImageBuffer);
    freeImageBuffer(outImageBuffer);
} else { // ONLY TWO ALGORITHM TYPES
    //ERROR! SHOULD NOT HAPPEN!
}

if (defInst->logFile) {
    CLOSE THE LOGFILE
}
```

Notice that we first check if the user wants the log file to be written. If so, it is opened at the beginning of SLIPprocess and closed at the end. Between these two statements, all fprintf statements should also be surrounded with the same if statement. This ensures that you are only writing to the file when the user has clicked the log file click box. Similar if statements exist for the ctrl_boolMakeNegative and ctrl_boolEnableBounds click boxes. These can be found in the code for pixel and plane algorithms within the cases for SLIPprocess. Also employed within these particular cases (omitted from Code Snippet 6) is the information collected from ctrl_floatFactor, ctrl_intValue and ctrl_floatMaskValues.

STEP 4: Attach callbacks to certain controls

From a data flow perspective, the plug-in is now complete. All the input data is collected and processed by the plug-in. But there is still some tidying up to be done. Specifically, we want to provide the GUI with a certain amount of intelligence: when certain buttons are pressed, only certain controls should be shown. For example, if the algorithm type pulldown menu is set to pixel, only pixel algorithms should be selectable. Likewise, if the apply factor algorithm is selected, the user should only be able to fill in the factor, but not a value to be added, nor mask values. We would also like to use some drawing functions to make a line underneath the folder tabs, thus creating the illusion that clicking a tab changes the lower portion of the panel. The mechanisms by which all of this is achieved are callback functions, commonly referred to as callbacks. We want to associate a callback function with certain controls, so that clicking them will result in immediate changes to the GUI itself.

In the callback version of slip, slipCallback.c, the following controls have been equipped with callbacks:

ctrl_butSave, ctrl_butLoad, ctrl_butAbout, ctrl_popAlgType, ctrl_popPixelAlg, ctrl_popPlaneAlg and ctrl_tab. The ctrl_butSave and ctrl_butLoad controls need callbacks (cback_butSave and cback_butLoad) so that when you click them, a requestor pops up allowing you to load or save your current settings. The ctrl_butAbout button uses cback_butAbout to paint an informational window proclaiming to the world that you authored the plug-in. The cback_popAlgType callback instructs the GUI to display either ctrl_popPixelAlg or ctrlPlaneAlg, depending on which type is selected in ctrl_popAlgType. Likewise, the callbacks for ctrl_popPixelAlg and ctrlPlaneAlg instruct the GUI to only show the controls relevant to the selected algorithm. Finally, the cback_tab displays only the controls associated with the selected tab. In order to make the tab look really good, we must also attach line drawing functions to both the tab (do_tabLines) and the panel (redraw) itself. Try commenting one or both of these callbacks out and see what happens. Here are some code snippets illustrating the use of callbacks:

Code Snippet 7: Code from createAlgCtrls to assign the function cback_popAlgType to ctrl_popAlgType.

```
CON_SETEVENT      (slipData->ctrl_popAlgType,
(int)cback_popAlgType, (int)slipData);
```

Code Snippet 8: This code, from cback_popAlgType, gets the value of ctrl_popAlgType and sets the visibility of ctrl_popPlaneAlgs and ctrl_popPixelAlgs accordingly.

```
GET_INT      (slipData->ctrl_popAlgType,      slipData-
>algType);
if (slipData->algType == ALGTYPES_PIXEL) {
ERASE_CON(slipData->ctrl_popPlaneAlgs);
RENDER_CON(slipData->ctrl_popPixelAlgs);
REDRAW_CON(slipData->ctrl_popPixelAlgs);
} else {
ERASE_CON(slipData->ctrl_popPixelAlgs);
RENDER_CON(slipData->ctrl_popPlaneAlgs);
REDRAW_CON(slipData->ctrl_popPlaneAlgs);
}
```

Once the callbacks are implemented, we can compress the GUI into a much more compact and efficient layout. This is illustrated in Figure 2 and Figure 3. Notice the smaller, more efficient use of screen real estate. The folder tabs allow us to stack the algorithm and settings panels, thus freeing up room for additional controls. The code for this callback-enabled version of SLIP is in the file slipCallback.c.

I hope you enjoy experimenting with SLIP as much as I did writing it. Like most code, SLIP is a work in progress. There are almost certainly quirks and bugs to be found and fixed. There are also improvements to be made, such as the addition of more image processing algorithms. On the GUI front, a preview tab might be nice as well. Even if you would rather not modify the code, you can still see evidence of the different ways a computer can do the same thing. For example, try rendering large frame sizes using the pixelwise and planewise versions of each algorithm, then compare the rendering time and resource consumption. You may be surprised at the differences! Another cool thing to try is using a chain of SLIPs (there are 4 slots for image processing plug-ins) to produce cool effects for your renderings. Feel free to drop me a line if you find any bugs, add any neat features or discover some non-obvious uses.

Rob Gougher is a software engineer for Nexpress Solutions LLC, a joint venture of Eastman Kodak and Heidelberg Druckmaschinen AG. He can be reached at www.robougher.com.

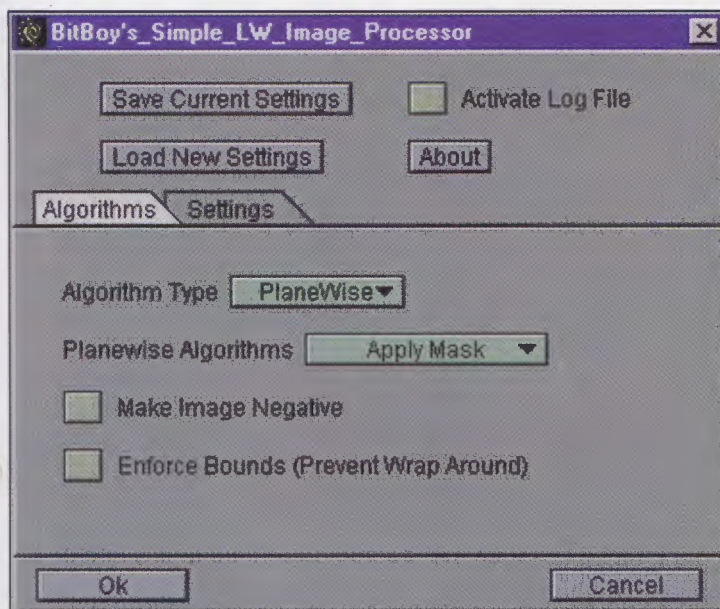


Figure 2: A screenshot of slipCallback.c. Note that only the algorithm controls are displayed when the algorithm tab is selected.

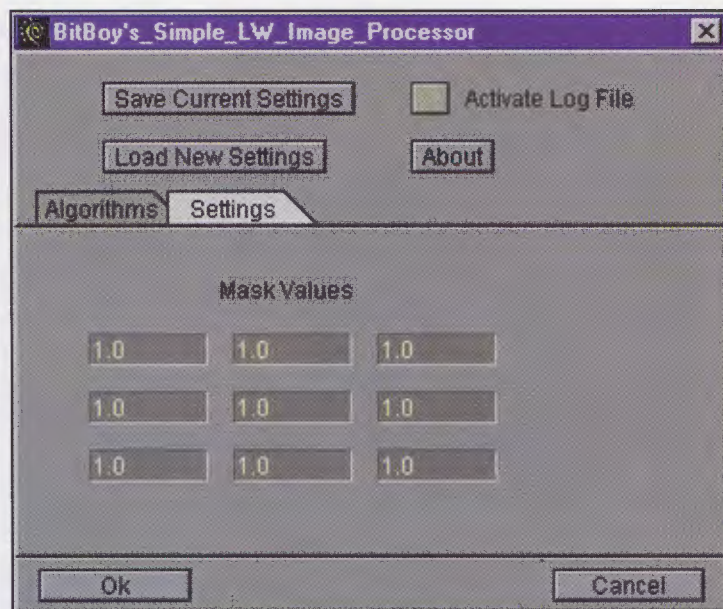


Figure 3: A screenshot of slipCallback.c with the Settings tab selected.

Heavy Metal

by Phil South

Like some really wise old Chinese guy once said, some things are really easy to learn but hard to master. I would certainly

number object surfacing among these pursuits, and let me tell you it comes pretty high on the list. And metals, well, they are some of the easiest things

to surface roughly, but the hardest to get really right. I have always loathed and detested the metals you get with LightWave, and as a part of this piece I will show you some of my versions of them.

It is very easy to reach for plug-ins to solve all your surfacing and modeling problems, but it is

better in some ways to see if you can get away with

standalone techniques first. It is a bit of beginner's cliché, but you will always find beginning anima-

tors complaining that "if only" they had such-and-such a plug-in, they could do this job really well. "It's a poor workman who blames his tools." Of course plug-ins are excellent and can improve productivity on more expensive productions, but they are often only as good as the person using them anyway. Anyone expecting plug-ins to wave a magic wand and make his work instantly better has another think coming. Plug-ins have their place, and used carefully and with style they can add a subtle patina of class over your surfaces.

In this tutorial I will cover standalone LightWave solutions, plus the kind of really professional effects you can get with a selection of the most popular plug-ins. I will look at making subtle use of specular, diffuse and reflectivity to make metals more realistic. I will skim by the Dosch plug-ins, in particular the excellent DirtyUp tool, and see how they can grunge up your metals and make them look really rusty. And finally, I will show how the excellent surface tools in Gaffer and RealiTools make metals really shine, almost literally.

By Hand Is Best

Saying that I hate all the preset surfaces isn't strictly true. I really like the horribly fake ultra-reflective chrome and gold textures. It always says LightWave version 1 to me. But we're not here for nostalgia; we're here for realism.

Make a ball. Mine is about 1.6 km for some reason. I know the idea of a ball that size is preposterous, but that's just the way it came out. Let's load a surface. How about a nasty chrome? Load the surface silver.srf from the LightWave CD onto a sphere. Render up a sample and you'll see it looks more like a weird round cloud. Obviously these kinds of surfaces are designed for use with more complex models and flying logos, where you need to see reflections moving across the surface as something moves.

To make a more credible metal out of this, bring the Diffuse up to about 35%, the Specular up to 100% and the Glossiness down to 16 and the reflection down to 12%. Now do a test render and you've got your "basic metal number 1" as shown in Figure 1. The blurry quality of the highlight gives you the feeling that the surface might be slightly too rough to reflect anything, and turning the reflections down means that if it does catch anything to reflect, the effect is so mild as to be almost nonexistent.

Bumpy chrome is another good one, as this provides real diffuse reflections by applying a small bump map pattern to the surface to genuinely break things up. Try a diffuse of 40%, a Specular of 72%, and a Reflectivity of 18%. Then in the bump map channel add some Fractal Bumps, and make the texture size 10m each in X, Y and Z (due to

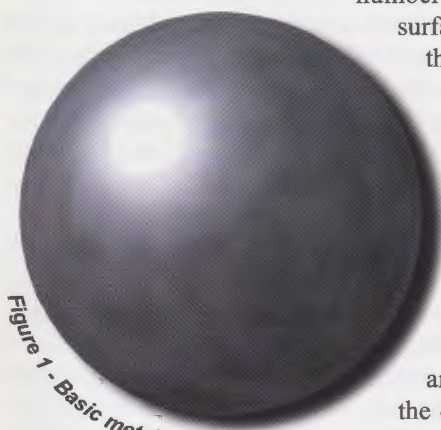


Figure 1 - Basic metal



Figure 2 - Bumpy chrome

the huge size of the ball). Make the surface color 200,200,200. Now render that out. Your results should look like Figure 2.

For a little more realism, you need to start detailing the surface. Try a Specularity map. This will put patches over the surface that varies in shininess depending on the color of the map. In this case lets try Fractal Noise. Turn off the Bump Map by holding the shift key and clicking the button (this is a handy keyboard shortcut for all buttons in LightWave). Instead click on the T button by Specularity, and chose Fractal Noise under Texture Type. What with the ball being too big, I had to enlarge the scale of the texture by making the Texture Size 50m in all dimensions. I also made the Contrast up to 3.0 to make the Fractal Patches a little more noticeable.

Another way to disturb the surface and make it look still more real is to use a diffuse map. Also you could think about making a very fine set of criss-crossing lines in Photoshop or on a scanned in piece of paper. These could be applied as a cylindrical bump map to imitate the fine scratches you often get on the surface of a metallic object. Just scribble back and forth on paper and scan it in. Instant scratch map!

Dosch Makes Rusty Crap Easy

DirtyUp from Dosch Designs (www.dosch3d.com) takes your metals and makes them look like rotten rusty piles of crap, but in a good way! The plug-in uses procedural textures and adds really nasty rust and films of grunge in a way previously impossible.

It's all very easy to use, really. Once it is installed, you activate the plug-in on the Surfaces Panel. You may have to hit the Advanced Options tab to see the plug-in slots. Click into Options once you've chosen DirtyUp in the top slot, and you are sent into the Panel of Doom, Figure 3. At this point in time I haven't had much chance to play around with the variables in DirtyUp, and as you can see from the panel in Figure 3, it's a little daunting. There are a lot of variables and a lot of "slide it, render, look at it, slide it, render..." cycles to go through to learn how to use it better. For the moment though, and for our purposes, tweaking the presets will more than suffice to give us the idea of how really powerful this plug-in is.

Let's try one for size. Add the Dosch 3D DirtyUp plug-in and click Options. On the top panel of DirtyUp you can see the buttons

Load All and Save All. This is a kind of preset load and save deal, as it saves and restores all the settings from a particular surface.

Click Load All and go for Rust 1. The size of our ball is still preposterous, so once you've loaded Rust 1, select the scale on the front panel and change it to 50 meters. Obviously if I had chosen to model a more sensibly sized ball, I would possibly need to resize the scale to match, say 0.5m or even 0.05m. Let's render and see where it gets us. What we have here, Figure 4, is clearly some kind of iron cannonball that has been painted with some kind of silver paint. Lovely rust, and so easy to do.



Figure 4 - DirtyUp Rust



Figure 5 - DirtyUp Verdigris

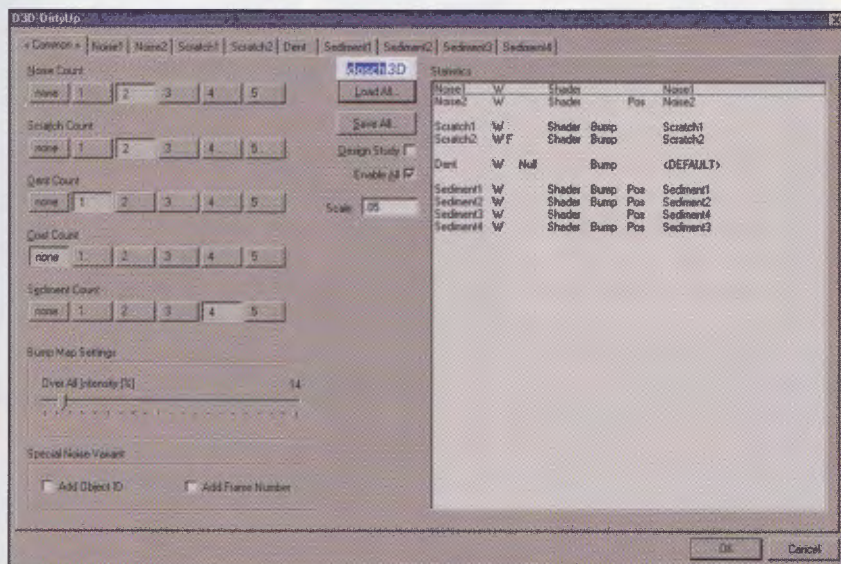


Figure 3: Dosch DirtyUp Options panel.

Okay, not an obvious one this, but take a common or garden-variety 3D cow, and use the DirtyUp plug-in with the verdigris preset. The cow is not as huge as the ball was, so a scale of a mere 0.5m will be enough. Now we have a marvelous corroded copper effect, Figure 5, perfect for the dome-shaped roofs of your Star Wars: Episode 1 cities. Nothing makes a statue look quite so lifelike as corrosion. Of course ramping up the lighting to 115% didn't hurt either, as this better replicates sunlight.

KEYFRAME MAGAZINE

Tutorials For
The Real World

Reviews You
Can Use

Subscribe
Today!

www.keyframemag.com

1-888-778-WAVE



Reality Tools

The excellent surface tools in Gaffer (www.Worley.com) and RealiTools (www.dynamic-realities.com) make metals really shine, almost literally. My favorite of the Gaffer set is Bloom, which makes the light carom off the surface of metal like a ricocheting bullet. In natural cinematography, light blooms out or bleeds on the highlights of shiny objects, due to tiny imperfections on the surface of the metal and the way that light or lenses interpret bright highlights. Often, a highlight on really shiny metal will start to offset the different wavelengths of light, giving you a blue, red and green cast. Either way, how LightWave does highlights isn't at all natural. By increasing the lighting on a scene by ramping the light levels up above 100%, you can start to bleed out the highlights, and this is more than adequate for most tasks. Also in Gaffer we have Anisotropic Specularity. This helps the realistic shading of surfaces like scratched or polished metals, hair, or threads, where the surface has a very complex specular behavior due to the fine detail on their surface highlights, where each has a precise dot with a kind a blurred halo around it. It's very hard to describe but very pretty to see, and makes a surface look finely detailed and very real.

By the same token, RealiTools is a fine set of tools for a variety of very specific tasks. Among those that can help you boost the quality of your metals are Fresnel, Lighter and Soft Reflections. I think Soft Reflections speaks for itself, with reflections in very shiny metals blurring and tapering off more realistically. Fresnel gives metals a different look altogether, giving the impression that the surface is very shiny and yet very finely scratched or pitted. Lighter is the icing on the cake and the tool that needs the most care. As I've said many times before, it's in the cinematography that most metals come alive, and this is true when lighting real metal as well as virtual metal.

Life Class For Lighters

So there are a number of very

good strategies for making metals look more realistic in LightWave. Of course, as I said before, plug-ins are not the be-all and end-all of realistic surfaces. They certainly save time and allow you to add spray-on class to a scene. But nothing really sells a shot quite as much as a good eye for lighting and shading. If you can look at a surface in real life, almost like life drawing, and then reproduce that in the LightWave camera, then you've gone beyond mere 3D hacking. You've entered an area of artistry. I thoroughly recommend actually doing this: Take a metal object, a Zippo lighter or another metal object from the home, and examine it through a magnifying lens. Examine its contours and its color. Examine the surface minutely. What makes it look that way? Does it have tiny surface scratches? Does it have well-defined highlights or are they blurred? Is it one highlight or does it have two overlapping? Get into the deepest layers of the detail on the object and find out what makes it tick. Then using the object as reference, build the object in LightWave and surface it. Remember that at least half the realistic look of any surface is in the lighting and framing of the shot. Pay close attention to the cinematography before you start complaining you can't get it right.

If you can reproduce a real object right away then fine, you are a genius and nothing will stand in your way. More likely you will make something similar to reality or even very far away from it. Don't worry. If you were painting the thing with brushes and paint then you'd have the same trouble. It takes practice to observe and reproduce, in any art form. Don't give up if you don't get it right the first time, because you will be surprised how little effort you have to make to learn and grow.

Phil South is a freelance animator and web designer based in the West Country of England. He can be reached via his web site on: www.ii3d.freemove.co.uk.



UVview 2.1 - 3D utility used to texture map objects parametrically.....**475.00**
UVview Shader Plug-in - Reads the UV file to render the UV mapped object in LightWave.....**71.99**

BlowUp & Shatter For LightWave
 Shatter any object; use BlowUp to animate the pieces for realistic explosions.....**89.99**

Power Plug-ins For LightWave, Intel
 Includes BlowUp, Shatter, Swarm and more, to enhance your objects...**209.99**

ReelMotion 1.0 - A stand alone program on CD. Can be used with any animation program that supports Biovision or Acclaim motion capture file formats.



ReelMotion is a realtime simulator that uses physics and collision detection to create realistic animations of objects and vehicles including cars, aircraft, motorcycles, helicopters and nearly any rigid body object.....**634.99**

Motion Designer 2

The powerful physical simulation plug-in for LightWave is now available in Ver. 2! Provides new functions such as self-collision, collision among multiple objects, and more. Motion Designer 2 is highly effective in creating animation of objects such as cloth.....**399.99**



Motion Gang
 Hitachi's motion plug-in for LightWave allows you to animate human characters by specifying the parameters.....**229.00**

PolygonEater

Reduce the number of polygons while maintaining the characteristics and shapes of models.....**159.99**



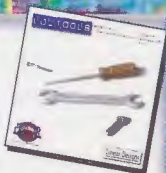
PGL - The plug-in that applies the bone deformation, different from that of LightWave 3D.....**189.99**

© STORM, Inc.

YOUR NUMBER 1 SOURCE FOR DIGITAL VIDEO, ANIMATION, MULTIMEDIA AND WEB DEVELOPMENT

Inspire 3D/LD_Tools Bundle

SAVE BIG on this bundle! LD_Tools gives Inspire 3D many numeric functions like the ones found in LightWave 3D.....**429.99**
LD_TOOLS.....**79.99**



Guaranteed to be in stock! Call now!

SURFACESROM

SurfacesROM

A collection of 300 Seamless Building Material Textures in

low, medium and high resolution files. Each texture is perfect 3D for mapping with any 3D program, and most include Bump, Reflection & Specular maps. Categories include Marbles, Stones, Grounds, Woods, Parquet Floors, Pavements, Metals, Walls, Roofs, Plants, and more.....**45.00**

Dinosaurs ROM.....**119.00**



LIGHTWAVE PLUG-IN/ LIGHTWAVE RELATED SOFTWARE

520 Photoreal Seamless Textures.....**37.99**
 Amazing Sci-Fi Textures.....**129.99**
 Amorphium.....**129.99**
 Animation Master 99.....**176.99**
 Aqua Pack.....**179.00**
 AuraFX Plug-ins.....**Call**
 Bryce 4.....**193.99**
 DarkTree Textures.....**289.00**
 DirtyUp.....**249.00**
 Enlightenment (electronic version).....**79.00**
 Fiber Factory 2.....**119.99**
 Fire CD.....**38.99**
 Genesis V2 Pro.....**295.00**
 Grime.....**Call**
 Lip Service.....**289.00**
 LW 3D-ROM.....**49.00**
 MacroForm 2.0, Intel/Mac.....**289.00**
 NatureFX.....**699.00**
 Particle Storm 2.0.....**489.00**
 ProPack One.....**149.00**
 Puppet Master.....**299.00**
 Pyromania Pro, Intel.....**169.99**
 Pyromania Set, Intel.....**155.99**
 RealTools2.....**249.00**
 Relativity.....**195.00**
 Seamless Textures 1,2,3.....**Call**
 Space Wars Digital Effects Kit.....**119.99**
 Tree Druid, Intel/Mac.....**169.00**
 Visual Effects CD 1&2.....**159.00**
 Visual Medicine.....**109.00**
 World Construction Set 4.5 Intel/Mac.....**629.99**

NOW SHIPPING!

Video Toaster NT - Never again compromise the quality of your video - lossless video editing is now available on your desktop! Introducing Video Toaster NT - the next generation, uncompressed digital video solution from NewTek. For only \$2995, your complete post-production studio includes: • Speed Razor (award-winning non-linear editing) • Aura (2D graphics, manipulation & compositing) • Lightwave (professional 3D animation) Already own a Video Toaster, Flyer or Lightwave? Save \$1000 when you upgrade to the Video Toaster NT! Complete reliable, quality turnkey Toaster workstations are also available!



LightWave 3D 6.0.....**2495.00**
LightWave 3D Upgrades.....**Call**
Aura.....**Call**
Aura Complete - Includes NewTek's Aura software, two tutorials Aura Video and Aura Convergence, and AuraFX Volume 1 plug-in.....**699.99**
Graphics Studio - Includes LightWave 5.6, Aura, Motion Pack, and Film Grain, plus Sound Forge XP from Sonic Foundry.....**Call**
Inspire 3D - THE 3D modeling, and rendering animation program for graphic designers, multimedia developers, and web designers who don't want to pay a high price to create, render, and animate 3D projects.....**Call**

For more information:
www.videotoasternt.com

Life Forms 3.5

Life Forms 3.5 LE The latest version of the defacto standard for creating realistic figure movement for 3D animation. Its intuitive interface and powerful features will help you save time and be more productive than ever. Our new entry-level package gives you all the features of Studio without the application support of 3D content. Great for creating Quicktime movies, VRML output for the Web, input for Strata Studio or trueSpace.....**179.00**
Life Forms Studio 3.5 - All the great features of LE, plus support for many of

today's top 3D animation packages. The Studio version includes PowerMoves 1 & 2, a collection of over 750 motion files (valued at up to \$5,000) to get you moving fast!.....**449.00**
Power Models - A collection of over 40 ready-to-use models, providing a firm foundation for your character animation needs. Populate 3D projects with diverse characters, and create multiple character.....**149.00**



Oxygen GVX1

A 100% geometry and lighting acceleration card which integrates the GLINT R3, and the GLINT Gamma G1 processor on a single AGP board.....**Call**



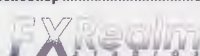
The VX1 uses the GLINT R3 processor to achieve true 2D and 3D superb graphics. In addition, it comes with 3Dlabs' Pentium III-optimized OpenGL drivers for strength and performance.....**Call**

Oxygen ACX, AGP.....**Call**
Oxygen GMX.....**Call**
Oxygen RPM.....**Call**



Creature Creator for LightWave
 This CD allows you to make one-of-a-kind characters in minutes. Also, included is a library of body parts, which can be tweaked or combined in endless ways for original creations, and an extra plug-in as well.....**225.00**

Creature Creator - New Standalone version! Can be used with any 3D software that imports the .3ds format.....**189.00**
Creature Creator trueSpace4.....**189.00**
MetaGroove 1.0.....**49.99**
SceneShop.....**39.00**



Homes: PDA accepted from schools and government agencies • Personal & Business checks accepted on CODs • Defective products replaced promptly. RMA number required (call 414-548-8159) for all merchandise returns. Returns not accepted after 15 days. Returned products must be in original packaging. No postage needed. Opened software not returnable. Shipping charges not refundable. Returns subject to 15% restocking fee. Not responsible for delays. Prices subject to change.



Information: 414-548-8120
 Fax 414-548-8130

Tech Support / RMAs:
 414-548-8159, 12-4 pm Mon.-Fri.

Safe Harbor Computers
 W226 N900 Eastmound Dr.,
 Waukesha, WI 53186



ORDER ON-LINE
 24 HRS. A DAY

To Order Call 800-544-6599

MON. - FRI. 9-5:30 CST

www.sharbor.com



Faking a Wake



It seems like such an easy thing to have a boat create a wake across the surface of a body of water. It looks easy, but putting it into practice is a bit trickier...

The problem is that the surface ripples need to be connected to the motion of the boat. There is also the problem of a partially submerged boat. If you can see inside the boat where it dips below the surface of the water, the polygons of the water will appear inside the boat. These are the types of problems that you never really consider until someone actually wants you to do it.

With the advent of reference objects, it is possible to tie the motion of one object to a texture attribute on the surface of another object. By jumping through a few hoops we can sail a boat that produces a wake as it moves across the water surface

and keep the inside dry through the creative use of reference objects. In this tutorial we will use very basic objects to illustrate this technique. It is important to use the values given here to get the same results. Later you can adapt the technique to your personal projects.

Make a Lake for the Wake

The first step is to make a water surface object. We need a 20-meter flat plane that is segmented 100 times on both the x and the y-axis. In modeler select the make box function and numerically enter these values: Low x=-10, y=0, z=-10, High x=10, y=0,

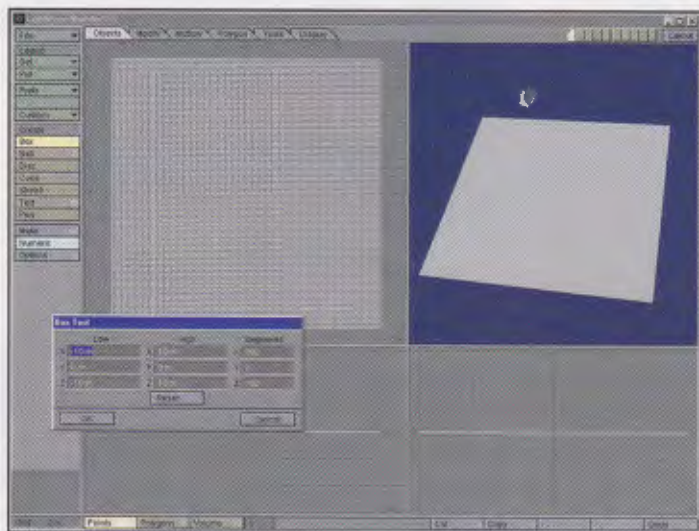
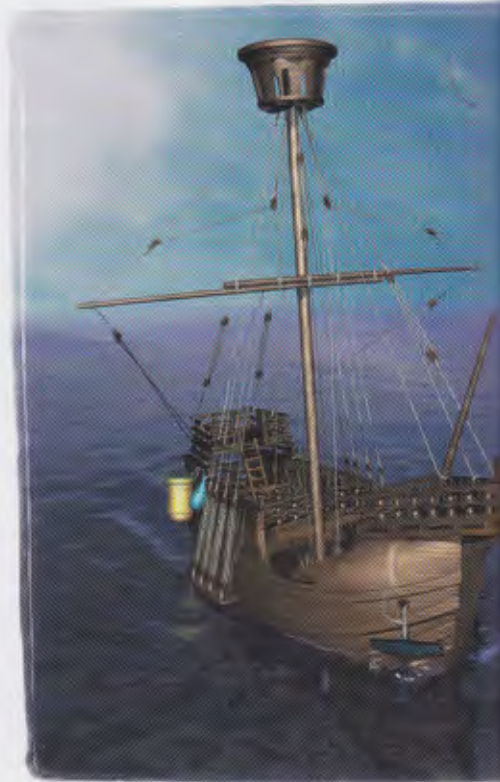


Figure 1: The modeling of the subdivided water surface object.

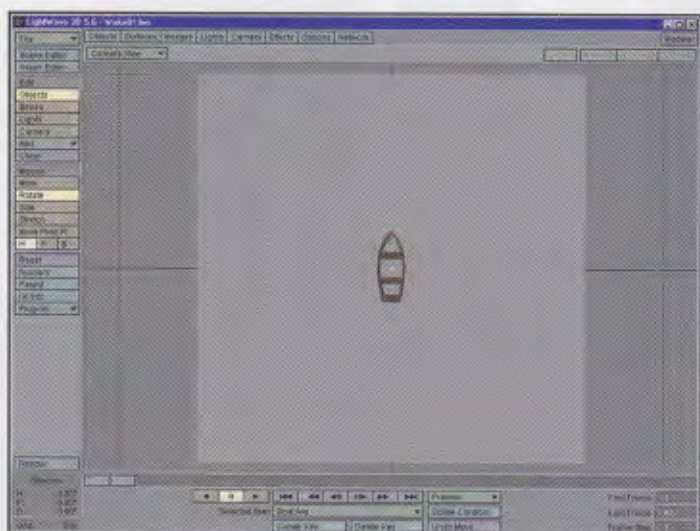


Figure 2: Overhead view of the boat centered on the water object. The limited region indicates the area used to create the clip map inside the boat.

$z=10$, and Segments $x=100$, $y=0$, $z=100$. Click on OK and press Enter to create the big flat segmented polygon. The normals need to be facing up and 3-pointed, so flip and triple as necessary. Finally, name the surface 'Lake' and save the object as Lake.lwo, Figure 1. This is the only object we will be creating specifically for this tutorial.

Layout at the Lake

We need to do a couple of things at once here. Load the Lake.lwo object and the stock rowboat object. The rowboat object can be found in the Objects/vehicles/marine directory. This is a crude rowboat-shaped object about 6 meters long and 2 meters wide. You can substitute another boat object of similar scale if you like. We need to create two maps with data we grab here. First, we need an overhead view of the boat resting on the surface of the water with lots of space around the boat, as in Figure 2. It is in the surrounding space that you will draw the wakes and rivulets. The shape of the boat is only there for reference and will be painted over. You can do a screen grab of the overhead view centered on the boat object, or position the camera directly over the boat and render out a single frame. Call this image 'Wake'.

A second image is almost identical, except that we want only the boat and where the water penetrates inside the boat. We need this image to be exactly the size of the whole boat as viewed from above. You can render it with the camera using limited region or do a screen grab and crop it later. Call this image 'WaterClip'.

Load the Wake image into your favorite paint program. Look at the overhead view of the boat on the water and imagine how the water would be pushed aside as the boat moves along. We want the top of the waves to appear white and the rest of the image to fade from gray to completely black at the edge. Use whatever tool you need to paint, smudge, or airbrush the waves to look like Figure 3.



Figure 3: This image was created using the overhead view of the boat on the water as a reference.

The second image needs to be cropped so that it contains only the top and sides of the rowboat. The inside of the boat where the water appears should be painted completely white. The sides and outside of the boat should be completely black. Save this black and white image as 'WaterClip'.

Hooking Everything Together

Go back into Layout with the lake and rowboat objects loaded. Load the Wake and WaterClip images. We will eventually be applying the Wake image as a displacement map. However, in order to get the scaling data we need, it is much easier to apply it first as a color map. This enables us to scale it interactively using OpenGL and real-time image mapping.

In the Surfaces panel, select the 'Lake' surface. Click on the T button for the surface color to apply an image map. Make sure Texture Type is Planar Image Map. Select 'Wake' as the Texture image. Turn off Width Repeat and Height Repeat. Turn on World Coordinates. Click on Add Reference Object. Important! Leave Texture Size, Center, and all the other settings at the defaults. Switch the Texture axis to Y. Click on Use Texture and return to Layout.

Before you do anything else, parent the ref_null to the boat object. If you have real-time texture mapping enabled you should see the Wake image roughly mapped onto the surface of the Lake. As you move the boat object the Wake image should follow along. It may be necessary to position and scale the Wake image to match the boat, so select the ref_null and keyframe it at zero with the proper values. When you are satisfied with the location and the scale of the wake, return to the Surfaces panel and remove the image map as a color texture.

To make a subtle wake, go in to the bump map texture panel and apply the wake image exactly as you did when you applied as a color map, but instead of adding a new texture object, use the one that is already there. That reference object has already been positioned and scaled with the assistance of the real-time mapping function, which is available to color mapping but not to bump mapping.

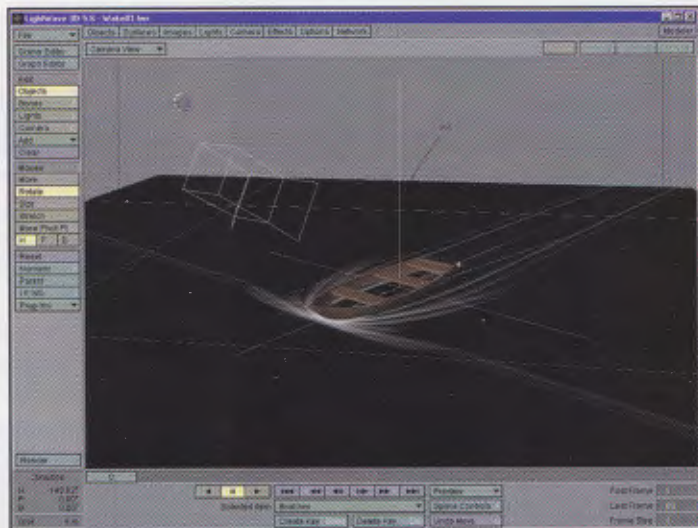


Figure 4: The ripple texture map is applied as a color map for the purpose of determining position and scale using the reference object.

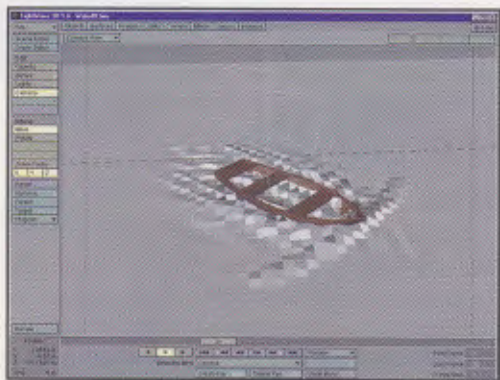


Figure 5: The image map of the ripples are removed as a color map and applied as a displacement map. The water surface could use some smoothing.

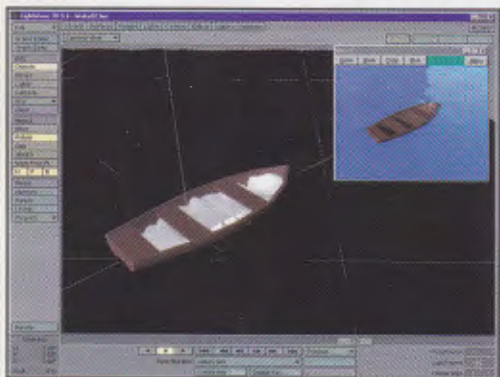


Figure 6: A clip map is applied to the surface of the water to remove the appearance of the water inside the boat. The boat shape is applied temporarily as an image map to determine the position of the map.



Figure 7: Layout view of an actual application of using reference nulls for ripple and displacement map textures.

For a more severe wake, we can apply the same image as a displacement map. Go into the objects panel and click on the texture button for displacement mapping. Apply the wake image as a displacement map exactly the same way as previously described, but still using the original ref_object. I suggest starting with

texture amplitude of 0.25, which should be about right with the values we are using here.

A Little Theory

I wish I could take credit for being smart for the things I discover by accident. When a texture is using World Coordinates, it is in effect un-parenting the texture from the surface. If the object with the surface moves, the texture stays put. By giving that un-parented texture a reference object, you are parenting to that object.

In the situation we have set up here, it is possible to move the lake object and the wake will still be connected with the boat. Depending on the speed of your computer, you may be able to see the lake object being displaced in real time as the boat moves along. This is as far as we are going with a displacement map for now, but if you are feeling really brave try adding a second displacement texture of ripples that also is using world coordinates, lots of falloff, and is parented to the reference null.

In the next step we will be using the same technique to position a clip map on the lake surface in order to keep the water out of the boat.

Keep Your Feet Dry

This is very similar to before except for the sizing. What we need to do is scale the image to the exact area the boat covers. To do this we automatically size to the boat, then apply those values to the lake. In the Surfaces panel, select the boat surface and click on any of the texture buttons (we will be removing the texture when we get the values).

Use planar mapping on the Y-axis and select the WaterClip image. Click on automatic sizing. Write down all the values for texture size and texture center including the proper units. When you have all the values, click on Remove Texture.

In the Objects panel, select the lake object and go to the Clip Map texture button. Select the planar map on the Y-axis and turn off the width and height

repeat. Instead of using automatic sizing, which would scale the image to cover the whole lake, enter the values for texture size and texture center that were calculated when you applied the image to the boat surface.

Once again turn on world coordinates, but this time use the boat itself as the reference object. Even though the clip map is being applied to the Lake, we have positioned and scaled the map to match the shape of the boat where it intersects the water. Because the boat is being used as a reference object and world coordinates are turned on, the clip map clips out the water wherever the boat is. If bits of the water are still appearing inside the boat, it may be necessary to adjust the centering and scaling values of the map.

Going Further

This tutorial was intended to explain how to use world coordinates with reference objects as sort of free floating texture effectors. The same technique could be applied to a character running fingers through sand or a wall that is glowing red hot from exposure to a lens flare.

To get an even better wake effect I often combine several displacements. Try putting a second reference null slightly higher and ahead of the boat with a ripple texture and a high amount of fall off on the Y-axis, so when the front of the boat dips down bigger ripples appear. You can also tilt the reference null forward so that the ripples swell out like a wake.

This covers most of the geometry of making your surface react like water. Texturing your water surface is a job for Fresnel shaders and a whole new tutorial. It is great to discover that there are still things left to discover in the way we use LightWave.

Pat Beck is the owner of Electric Crayon Studios in Milwaukee, WI. He has produced animations for Harley-Davidson, The Milwaukee Wave, and other local companies. You can email him at zippie@execpc.com.

Texture & 3D Model Libraries for Lightwave 3D

LIGHT-ROM 8

All new collection of Lightwave models for \$39.95



"Ant" by
James Curtis
www.ncweb
.com
/biz/jmcfx



"Fairy" by
Alex
Lugones
www.
wildseed3d.
com

"Opera" by
Frank
Maturi
www.wild
dog1.com/
frank.htm



"Frog" by
Don Drennan
http://home.
sprynet.com/
~ddrennan/
home.htm

Loren Sharp
www.geo
cities.com/
capecana
veral/
hanger/6004



"Peugeot"
by Ingo
Haverich
Independ
entarts@T
-Online.de

"Puffer" by
www.burn
ingham
studios.
com



LIGHT-ROM GOLD, 4, 5, 6, 7 & 8 Super Bundle only \$79.95 (all L[6] compatible)

SURFACESROM

**SurfacesROM
Only \$49.95**

300 seamless Building Material textures designed for 3D mapping by Freeminds3D. Includes Bump & Specular maps. Includes Marbles, Stones, Roofs, Metals, Woods, Floors, etc.



SurfacesROM, DinosaursROM & 520 Photoreal Seamless Textures Bundle only \$99



**LIGHT-ROM 7
Only \$39.95**

Collection of 100s of royalty free Lightwave models licensed from Syndesis Corporation, includes Digital Elevation Maps (DEMs) covering all the land on Earth!



All CD-ROMs work on all platforms and are L[6] compatible



**520 Photoreal Seamless
Textures Only \$39.95**

520 seamless textures designed for 3D mapping by Kevin Tyler Enterprises. Includes Buildings, Doors, Vehicles, etc.



Either SurfacesROM or DinoROM & 520 Photoreal Seamless Textures only \$69.95



DinosaursROM

Reg. \$125, now \$49.95

10 Dinosaur Lightwave 3D objects created by Freeminds3D with IK. Also for 3D Studio Max.



Graphic Detail Inc., 4556 South 3rd Street, Louisville, Ky. 40214 USA
www.lightrom.com ~ 1-800-265-4041 ~ 502-363-2986 (voice & fax)
Mastercard & Visa accepted ~ Shipping is \$4.95 in U.S., Canada & Mexico

LIGHT-ROM is a trademark of Graphic Detail, all other trademarks are the properties of their respective owners.



Reviewed this Month:

Software

LipService

Amorphium

Hardware

Graphics Chipsets

Video

Masters of Visual Effects

LipService

\$299 Intel (WinNT)

Joe Alter, Inc.

www.joealter.com

reviewed by Doug Kelly



I've been evaluating LipService for a couple of months now, and I have mixed feelings about it. On the good side, there are some very powerful tools here, and the interface for some of the tools is very intuitive and a significant improvement over other lip sync tools I've used. On the other hand, the program has a very rough interface design. It appears to be a late beta test version, and needs quite a bit of work before I would call it a finished release. I can't honestly recommend it to beginning animators, primarily due to the excessively steep learning curve. If you are an experienced animator, and are familiar with any other lip sync software (or even the manual dope sheet approach), you may be able to figure out how to use LipService effectively.

In evaluating any production tool, the most important factor is the quality of the final product. LipService has produced (in expert hands) some of the most entertaining lip sync performances I've seen in the 3D CGI medium. If you are considering LipService, I highly recommend that you view the animations on this issue's CD-ROM, or download them from Joe Alter's website.

LipService is a fairly successful attempt to reconcile the dichotomy that is

part of the lip sync process. On the one hand, lip sync key poses are dictated by the voice track, with very little room for the animator to interpret how long or how strongly to emphasize a phoneme. This requires a rather technical, analytical approach. On the other hand, the facial animation to express emotions is highly subjective, and the animator has a great deal of freedom (and responsibility) to create a strong performance. This requires a fluid, improvisational approach. The problem is how to successfully support both these approaches in the same software.

LipService is a standalone application plus a LightWave displacement map plug-in. You perform the lip sync operations in the application, then import objects and deformation files into LightWave for rendering. I believe this approach gives the program a pretty good chance of working with LightWave 6, but no one has promised anything.

LipService requires a set of morph targets with identical point count and order, the same rules as for Morph Gizmo or MTSE. The tutorial files include the base Zeke character face, 18 phoneme targets, 32 facial expression targets, some scene files and a WAV voice track. The models are very well done, and it really helps the new user to start using the program with such a nice data set.

The work area (the large window) can show the loaded object as wireframe or shaded solid, and allows you to make surfaces appear white, green or invisible. White means you can manipulate the surface, green or invisible means the surface is frozen.

If you need to create a set of morph targets, LipService includes some unique and powerful tools to help. You can load the base object into the work area using the GET: MODEL menu option, make changes to it with a variety of sculpting tools, then save the modified object to one of the Pose slots. When you have completed sculpting your morph targets, the PUT: POSES function saves all 50 objects in a numbered sequence for easy reloading.

One function that I especially appreciate is EDIT: DEFORM POSES. This measures the changes you have made to the work area (presumably the base or neutral object) and applies those changes to all 50

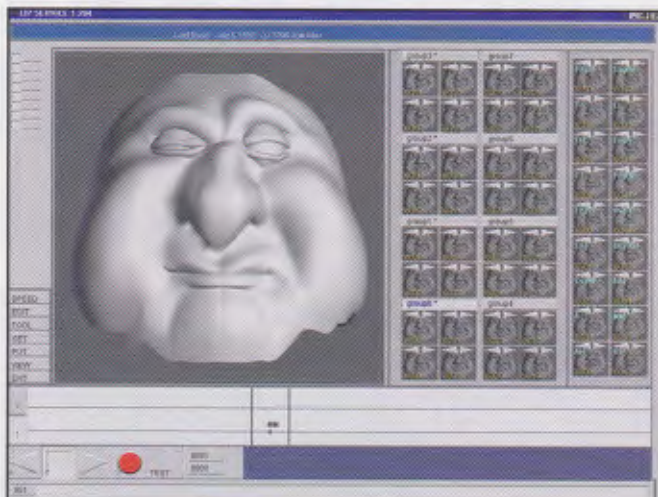


Figure 1: LipService interface.

target objects. This can be a lifesaver if you have a lot of finished facial animation and the client wants to make changes to the character's face. This function enables you to make changes to the character right up to the last render. If you make a living with facial animation, the Deform Poses function alone may justify the price of LipService.

As in LightWave's Modeler preview window, you can rotate the object in the work area by holding down the left mouse button and dragging. The sculpt tool is activated by holding down the right mouse button. You can vary the range of the sculpting tools by holding down both mouse buttons and dragging out a circle to cover the area you want to affect. The sculpting tools include translate, inflate and to neutral. Translate moves vertices in the XY plane, parallel to the screen. Inflate moves vertices in or out along their own normal axes, which is very useful for organic swelling or sucking-in changes. To neutral is like a controlled Undo, it changes the current object back toward the shape of the base or neutral object. This is especially handy if you have over-exaggerated a change and simply want to reduce it a little.

Once you have a set of phoneme objects, you can load a WAV file and start setting keyframes. You can drag the frame slider with the mouse, or step frame-by-frame using the left and right arrow keys. For continuous playback, the reverse, stop, and play icon buttons are nonstandard but relatively easy to figure out. The keyboard shortcuts for them are, respectively, A, S and D. This keeps the most-used shortcuts under three fingers of your left hand, so once you get used to it you can operate these function very quickly. The audio playback is disabled in reverse, or with the frame slider; it only works in forward with the arrow keys or the play button.

You set a key on the current frame by left-clicking on the cell of the phoneme object. For example, I clicked on cell 43, labeled MM, to add the MM key to frame 1 of the first track of the animation, as shown in Figure 1. If you make a mistake, the backspace key deletes the current keyframe. You continue setting, shifting and deleting phoneme keys until you are satisfied with the lip sync. You can get a more accurate playback with the VIEW:

MAKE PREVIEW menu option, which creates a smaller animation synchronized to the WAV file.

Once you have the lip sync locked, you can select track 2, the performance track, and animate the rest of the facial expression. Instead of requiring you to select individual poses and set keyframes, LipService enables the user to record a live performance, making this part of the process very much like puppeteering. You select one of the performance groups, each of which includes four cells. These four cells should contain complementary poses; for example, up and down positions for both left and right eyebrows. You then simply move the mouse cursor over the work area to set the weighting of each cell from the group. For example, moving the mouse to the upper left corner of the work area sets the upper left object of the selected cell to 100% weighting; returning the mouse to the center of the work area reduces the weighting to 0%. To get used to this technique, LipService has a Test mode that allows you to play with the selected group's weighting without recording the performance. Once you are ready to record, you simply press X to start recording, move the mouse to create the performance, and press <ESC> to stop recording. If you don't like the results, you can press C or choose EDIT: CLEAR PERFORMANCES to erase the performance track. Once you get the hang of it, this is the slickest way I know of to layer a performance over lip sync.

My number one complaint is that this is a Windows application that doesn't look or act like any other Windows application I've ever seen in commercial release. Most of that is due to its origin as an SGI application written for Softimage users; if you use Soft, you'll feel more at home. However, people who use Windows are very much used to certain basic behaviors, and if a program doesn't look and feel the same way, users get confused and frustrated. In Alan Cooper's *About Face: The Essentials of User Interface Design*, one of the author's prime tenets of interface design is, "Don't make the user feel stupid." LipService made me feel stupid every few minutes for hours, until I'd pretty well mastered the program.

The LipService interface (Figure 1)

has no menus at the top, and the side menus don't look or behave like Windows or MacOS menus. The GET, PUT and EXIT menus should be combined into a standard File menu using standard shortcuts and terms like Open, Save, and Save As. EDIT should be an Edit menu, with the addition of standard Cut, Copy, Paste, and Delete functions. LipService suffers from a serious lack of keyboard shortcuts; for expert users, this means limping along with the slower mouse-through-the-menus. Again, this is a standard feature of both Windows and MacOS software, and is not difficult to implement. Joe Alter's stated reasons for omitting some shortcuts is that they can be dangerous if triggered accidentally and that they are not used very often. I understand his reasoning, even if I don't agree with him on every point. Also, the sculpt tools should not use both left and right mouse buttons as this leads to cramping and can aggravate carpal tunnel syndrome and other RMIs. The standard practice is to hold down the CTRL key along with the left mouse button to modify the action of the mouse.

The LipService documentation needs a lot of work. As of this writing, it is about the size of my average Character Shop column. That's OK for one tutorial, but definitely on the light side for a combined reference and tutorial manual. I was able to puzzle out most of the program's functions, but it was hard going. If I didn't already have a lot of background with other lip sync animation software, I would have been completely lost. Standard Windows Help would be a big improvement. Since the manual is already in Word format, it should simply be a matter of converting it to a Help file. I understand Joe Alter is looking for someone to rewrite the manual; I hope a new manual is available by the time you read this.

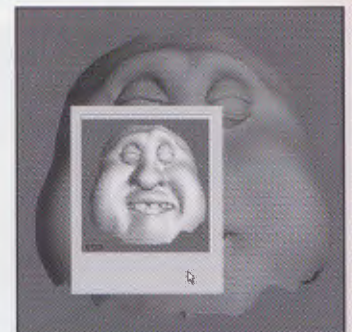


Figure 2: LipService preview animation.

Amorphium

\$150 Intel/Mac
Play, Inc.
www.play.com

reviewed by Jared Bendis

You know there is something special about a piece of software when it brings out a range of emotions in you. From the moment you see the box with its 3D prismatic images, you know that Amorphium is something different. Amorphium is Play's 3D real-time modeling and painting program, and at \$150 they have priced it within reach of amateurs and professionals alike.

The installation was very straightforward. I had no difficulties running it on any system from a 233MMX to a Pentium III 600. It didn't like my Sony Picture-book's 1024x480 resolution; Amorphium really needs that 800x600 minimum to run. But it didn't crash even when I tried to change video modes with the program running! Kudos to the software designers on outfoxing even my abuse.

The 454-page manual is thorough, and the tutorials do a good job of introducing users to the variety of tools available. However, the software is intuitive enough that I admittedly didn't need to use the manual much. Like most manuals on graphics software, I found that they tend to document how to use a feature much more than how to find the feature.

The interface is golden. With a model on the screen and with a few well-laid-out click and drags you can pivot, rotate, move, and scale the object. While the controls appear to make this happen, the real power is in Play's use of real-time

screen refresh without the requirements of special hardware. For those of us with accelerated chipsets it just zooms. You will notice a deviation from the standard Windows API, which I find a little distracting, but it is very forgivable.

Every model starts with a pre-loaded model, the Metaballs, or a primitive. There are many primitives to choose from including TrueType fonts. There is also a wide variety of special effects you can apply to the model. They are all worth playing with and you will soon find your own favorite. I particularly like to bend my model when it is rotated in various positions.

The modeling tools offer a variety of tips, though after much experimentation I found I almost always use the default tip. It took me a while to find the pressure setting to give it either a light touch or to turn it into a pull instead of a push. The first few times you use the software you are liable to push your way right through the model. It doesn't tear, it just creates a very funky spike way out into space. It's a good idea to keep rotating the model to see how you are doing. One of the more useful tools for character animators is the Symmetry tool, which enables you to make identical changes simultaneously to both sides of a model. This is extremely useful for making morph targets, since Amorphium also allows you to lock the point/polygon distribution to prevent illegal changes.

I enjoy using the painting tools. Pick your color and paint the whole model or just spray around into the details in the same way that you modeled.

Included in the software is a basic rendering engine. Since most professionals have their own rendering engine, this makes it clear that Play had the amateur in mind. The fog and smoke effects are very nice but only if the program is to be used for final output. Amorphium also has the ability to animate one item. Put this all together and you have the perfect software for quick and easy flying logos. This alone makes it worth the price of admission.

Saving is a little confusing; they really want you to work with a project. This saves your models to a project directory and

everything goes into it. If you try to save an object or an image without saving into a project first, your guess as to where the files end up is as good as mine. You should be working in a project type structure anyway so this isn't the end of the world.

I work with clay a lot, and am used to working with it to create my models. In addition, I just returned from SIGGRAPH where I had the fortune (or misfortune) of modeling for several hours on a force-feedback modeling system. Believe it or not, I like Amorphium better than the force-feedback system. Amorphium is much easier to use, the interface is actually usable, and after working on it for hours my hand is not paralyzed by crippling pain.

However, I still cannot make anything reasonably identifiable with this product. Having seen the results of others using Amorphium, my conclusion is that it is very much geared to the computer modeler and not the traditional clay modeler. The Metaballs function which allows you to join clay is not intuitive to the traditional modeler, but very much what the computer modeler is looking for in the rapid creation of basic shapes. The process is an additive one, small models are used to create larger models.

The reason for Amorphium's review in this magazine is that it purports to import and export LightWave objects. Unfortunately, this is not entirely accurate. Amorphium only imports triangulated models; if you have a quad polygon object - common for Metamotion cages and targets - Amorphium will automatically triple it on import. If you don't choose the right import settings, it will also modify the object's center point and drop some polygons along any open edges. If you carefully choose the same import settings each time, you will be able to get consistent, morphable exports. If you want to use Amorphium to modify Metaform objects, don't.

Overall I like it. I use it every day, but for me it is a toy first and a tool second, for others it will be a tool first and a toy second. Either way it is definitely worth getting.



Graphics Chipsets

reviewed by Julian Kain

Welcome once again to *Keyframe's* graphics chip roundup. This month we're going to take a slightly different approach to reviewing graphics products and their OpenGL performance in *LightWave*. Instead of looking strictly at chips, we're going to look at two new cards which represent the next generation of 3DLabs accelerators: the Oxygen VX1 and the Permedia 3 Create! These are very similar products, and will be compared side-by-side rather than the separated review format used in previous issues. Also, I'm adding another benchmark this month: the industry-standard OpenGL benchmark, *Viewperf*.

3DLabs' graphics products have always held a special place in the CG industry, and the Oxygen series of cards have been some of the best cards available. The Permedia 3 chip is the follow-up product to the extremely successful Permedia 2, which redefined consumer-priced, professional-quality graphics. To put these two cards into perspective, the Oxygen VX1 is the second card in the new generation of Oxygen cards, above the ACX but below the GVX1 and GMX in performance and price. It is designed to provide, as 3DLabs puts it, "industrial strength graphics for workstation professionals." The Permedia 3 Create! card is geared toward general graphics use and web design.

Benchmarking Information

The test system is a Pentium II 400MHz with 256MB RAM, using 16-bit color (unless otherwise mentioned). The Hummer Test, the main benchmark measuring *LightWave* OpenGL performance, is simple to duplicate: Load the Hummer scene from the directory "X:\NewTek\Vehicles\Hummer" or *LightWave* CD. In Layout Options, set the bounding box threshold to 56585. Make sure OpenGL is selected for Layout

View Graphics. Find a watch or clock to measure elapsed time, and click on the play button at the bottom of the screen. Allow the scene to play through frame 50 and stop timing. Note that your CPU does play a role in the speed you will see. However, increases and decreases in speed will be proportional among graphics chips, making this a valid benchmark. The Hummer Test relies heavily on your OpenGL hardware, and is geared mainly towards measuring polygon speed. Other benchmarks in this article include the X29 benchmark, *Quake 2 demo1*, and the *Viewperf 6.1.1* suite. *LightWave 5.6c* is used under both Windows NT 4.0 (SP5) and Windows 98 SE.

Card Information

3DLabs, Inc.
<http://www.3DLabs.com>
408.530.4700

Oxygen VX1 (based on the GLINT R3 chip)

Estimated street price: \$225

Hummer play time: 49 seconds in Windows NT, 53 seconds in Windows 98

Permedia 3 Create! (based on the Permedia 3 chip)

Estimated street price: \$175

Hummer play time: 54 seconds in Windows NT, 56 seconds in Windows 98

First Impressions and Setup

An extremely important aspect when dealing with new graphics hardware is support, or the ability to get both technical support and bug fixes from the manufacturer. Some companies are notorious for buggy drivers, and can be very difficult to work with or even to contact. I was extremely impressed with

3DLabs and their representatives from the beginning. This means a lot to me, as it should for any

buyer. 3DLabs is easy to contact, and a joy to work with. However, the level of

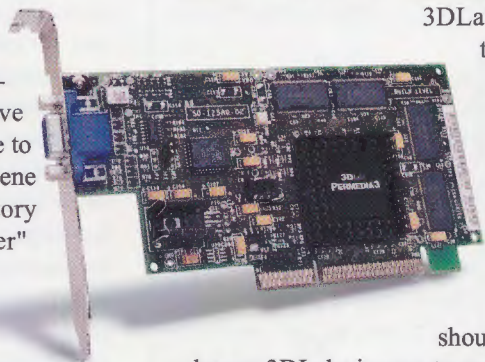
professionalism didn't end there. I was very impressed to see that the packaging of the cards included an anti-static wristband for installation and a Colorific color calibrator, along with a well-designed drivers CD.

Upon inspecting the cards, it became obvious how similar they are. Both cards are AGP (1x/2x) and use 32MB RAM. In hardware design, the cards look exactly the same, from the layout of the components to the smallest markings on the silicon, the VX1 and Create! are physically identical except for having different chips. The chips themselves are similar as well. I spoke briefly with Jean Salvati of 3DLabs who said that the Permedia 3 chipset (Oxygen VX1) are both based on the same design, though the GLINT R3 has been further optimized for OpenGL performance. More similarities include the drivers themselves; both cards use the same driver files, and upon custom driver installation (i.e. no InstallShield), you must pick which card you are actually using.

The similarities between the cards continue in 2D performance. Driver installation was flawless and both defaulted to 1024x768 display resolution on reboot. I immediately noticed the vivid, bright colors and absolutely razor sharp text that these cards produce. I had become accustomed to the rather murky looking TNT picture, which does display accurate colors but tends to be somewhat dark and to make text a bit too bold. The VX1 and Create! cards provided welcome changes, and overall, both cards performed well in 2D. Though these cards are not as fast as some of the newer consumer-level cards, 2D images were downright beautiful.

OpenGL Performance

3DLabs provides a driver applet (accessible via Display Properties in Control Panel) from which you can customize your OpenGL graphics subsystem. Included are gamma options, a VSync toggle, program-specific OpenGL optimizations (including *LightWave*) and other options. When considering the speed differences between the VX1 and Create!, it quickly became obvious how much the combination of



OpenGL-based hardware and LightWave 3D-based software optimizations matter. The Permedia 3 Create!, as similar as it had been to the Oxygen VX1 until the point of benchmarking, quickly showed its lack of OpenGL hardware optimization. In Windows 9x, the Create! is 6% slower than the VX1, and in NT it is 10% slower on the Hummer test. However, on the rest of the benchmarks the VX1 finds itself neck-and-neck with the Create!

So what "optimizations" are present in the VX1 that aren't in the Create!, and what separates the two cards in their LightWave performance? Not much. The GLINT R3 simply has a better rendering pipeline than the Permedia 3. It has been updated not only to be faster, but to include integrated rendering features such as filtered volumetric lighting. Coupled with the

LightWave-specific driver optimizations, this upgraded rendering pipeline puts the VX1 above the Create! in terms of LightWave performance. However, the Hummer test only shows polygon speed. What about the texture display speed? It turns out that the Permedia 3 and GLINT R3 chips are neck-and-neck in performance when a LightWave scene becomes heavy on the textures. This is because of a new technology in both cards, which 3DLabs calls Virtual Texturing. Most graphics cards, when storing textures, are limited to the amount of memory on the board. However, with Virtual Texturing, the VX1 and Create! can use up to 256MB of system memory to store textures. Furthermore, the technology uses a sophisticated non-stutter system to load (and display) only the sections which are needed, and not the entire texture.

All situations considered, the Create! averages a speed of between 5% and 15% less than the VX1. Though both cards (and especially the VX1) may seem unbeatable, comparing the cards to the tried and true TNT shows a different side of things. Using the Hummer test scene, the VX1 is only 6% faster than the TNT in NT and is actually 2% slower in Windows 98. The Create! is 4% slower than the TNT in NT and 8% slower in Windows 98. With the massive amounts of polygons the cards can handle, and the impressive optimizations of the VX1, how can a one year old, consumer level card compare? The answer lies in the system CPU.

Both the VX1 and the Create! use optional drivers dubbed "PowerThread". These drivers optimize OpenGL performance for processor-specific extensions, such as the Pentium III's KNI and AMD's 3DNow! They also take advantage of multi-processor systems. The testbed for this review is simply a Pentium II 400MHz machine, which has no special instruction set that the PowerThread drivers would benefit from (except for MMX, which really never did much anyway). As much potential as the cards have, they can be significantly held back by a lower-generation processor. However, with the right processor (upper-level Pentium III and AMD Athlon CPU's), they will fly, especially the Oxygen VX1.

Conclusion

Oxygen VX1: 9.5/10

Permedia 3 Create!: 8.5/10

Without a doubt, the Oxygen VX1 takes the crown this issue -- it's the fastest OpenGL card I've ever tested in LightWave 3D. It shows off excellent, trend-setting hardware features, and some incredible drivers. 2D is crisp and bright, and 3D can be very impressive. Also, as mentioned earlier, the card packaging is excellent, and support is top-notch. What more could you ask for? Well, maybe a faster CPU; if you don't have a Pentium III or Athlon, the VX1 won't perform as fast as it could. However, it's smart to take the optimistic view that you can grow with the VX1. It will increase in speed as your computer does. The same is true with the Create!, although it is simply not as fast as the VX1.

Hummer Scene		<u>Windows 98</u>	<u>WindowsNT</u>
(lower is better)	Oxygen VX1	53 sec.	49 sec.
	Permedia 3 Create!	56 sec.	54 sec.
	nVidia TNT-based card	52 sec.	52 sec.
X29			
(higher is better)	Oxygen VX1	197	277
	Permedia 3 Create!	205	273
	nVidia TNT-based card	278	254
Quake 2 Demo1			
(higher is better)	Oxygen VX1	35.6 fps	39.8 fps
	Permedia 3 Create!	35.4 fps	39.1 fps
	nVidia TNT-based card	37.7 fps	36.2 fps
Viewperf			
(higher is better)	Oxygen VX1		
	ProCDRS	N/A	5.9
	Data Explorer	N/A	9.4
	DesignReview	N/A	5.4
	Advanced Visualizer	N/A	12.2
	Lightscape VS	N/A	1.1
	Permedia 3 Create!		
	ProCDRS	N/A	6.1
	Data Explorer	N/A	9.4
	DesignReview	N/A	5.5
	Advanced Visualizer	N/A	12.1
	Lightscape VS	N/A	1.1
	nVidia TNT-based card		
	ProCDRS	N/A	3.7
	Data Explorer	N/A	8.6
	DesignReview	N/A	5.9
	Advanced Visualizer	N/A	8.9
	Lightscape VS	N/A	1.1

I hope you've enjoyed this installment of *Keyframe's* graphics chip roundup, and until next time, happy animating!

Julian Kain is currently a 17-year-old high school student who would love to hear your comments regarding this article! E-mail him at julian@juliankain.com or visit his web site at <http://www.juliankain.com>.

Masters of Visual Effects

Series One: Introduction to Visual FX
\$295 Four Tape Set
Puffin Designs
<http://www.puffindesigns.com>

reviewed by Lee Stranahan

Few colleges or universities have well-developed programs for training visual effects artists. Most visual effects artists I know are pretty much self-taught. It takes a combination of drive, intelligence, and on-the-job training, and there is usually a lot of trial by fire. There should be a better, more organized way to learn the craft of making fake images. That's where a new series of training videos comes in: *The Masters of Visual Effects* series, distributed by Puffin Designs.

Series One, a four tape set called *The Basics of Visual Effects* is out now. These tapes are not at all LightWave-specific; in fact, they do not demonstrate any specific 3D tools. Instead, this series is a broad introduction to visual effects as a whole. As such, it is valuable to every LightWave animator who wants to understand more about the visual effects process.

The tape's clever cartoon James Bond style intro lists about a dozen instructors, but this first installment of four tapes only features two of those instructors: Ron Brinkmann and Forest Key. The 'Masters' do have the professional credentials to back up that title. They are effects artists, largely based in the San Francisco bay area, almost all of whom seem to have connections to effects behemoth Industrial Light+Magic. Obviously, ties to ILM

mean work on projects like *Twister*, *Mission Impossible*, and the *Star Wars* films, as well as a number of high-end commercials.

There is a slightly cliquy feel to the tapes. The instructors obviously all know each other, and the tape has some inside references and jokes that are probably funnier to the people who made the tape than to viewers. On the other hand, the visual effects world is cliquy; maybe effects artists are making up for not being the cool kids in high school. By watching these tapes, you are getting a little closer to the inner circle of visual effects, and there is something to be said for that.

The series' production values are good, overall. The titles and graphics are well done, although they tend to be lists of things, words rather than images. Some of the lighting on the instructors is a little suspect; it is just not compellingly shot video, especially when compared to the quality of the graphics and the instructors. The tapes are well edited and paced, although there are a few gaffes that should have been fixed. On balance, the technical aspects of the tape get a B minus: good enough, but I expected a little better.

Tape 1, *Visual Effects Cues*, is hosted by Ron Brinkmann, whose feature film credits include *Contact*, *Speed*, and *The Last Action Hero*. Ron also wrote the book *The Art And Science Of Digital Compositing*. Ron seems a little uncomfortable in front of the camera. However, he does a nice job of explaining the elements that go into creating visual effects that fit seamlessly into a scene. The tapes make nice use of visuals to show how elements such as lighting and camera elements affect the viewer's perception.

The tape stresses that unless all the elements are just right, the viewer gets a subtle feeling that something is wrong, and that means you are not doing your job. This is the most basic tape in the series, but it is important material that really does need to be covered. Don't expect too many earth-shattering revelations here, but do expect to have a deeper and fuller understanding of what your job is in simulating reality.

Tape 2, *Film And Video As Digital Images*, seems designed to make your head explode. It is a discussion of the technical aspects of what makes up a digital image, and as such it is packed with numbers and math. In the wrong hands, this is stuff that could be deadly boring. Don't bother telling Forest Key that! He hosts this tape, and has worked on shows like *Twister* as well as heading up Puffin Designs, and he attacks this material with such relish and enthusiasm that you almost forget how dry the subject matter is. Forest really wants you to understand terms like '32-bit', and he's right; you should know and understand this material if you do visual effects. As an effects supervisor, you'll just look stupid if you don't understand the benefits of expanded color depth or embedded alpha channels. Even if you think you know this stuff, chances are you will learn a thing or three from Forest Key. His personality and presentation are dynamic enough to make you want to stick it through, even if you have to take a break somewhere to clear your aching head.

Tape 3, *Post Production Basics*, is also hosted by Forest Key, and is an overview of the post production process from the perspective of the visual effects artist. It covers both film and video, which is good, because there are some crucial differences between the processes. This is invaluable material, and it is especially important because it takes you through a process that is rarely seen by any one person.

This tape and Tape 4 both make use of a two-minute short film entitled *Keep Clear*. This film features a number of visual effects shots and looks good, although it does have something of a 'film school' feel to it. It also features a cameo by flamboyant San Francisco Mayor Willie Brown, and Da Mayor even does a bit of acting in front of a blue screen. Forest uses this short, which he worked on, for a number of examples.

There are also good hints on project organization and file naming. As boring as it sounds, good organization can really make or break a project, given the sheer volume of scenes, shots, elements, test renders, final renders, composites, and so

on. This is illustrated with an effects shot from *Keep Clear* that contains over fifteen different elements but fills only three seconds of screen time.

Tape 3 also does a very nice job of explaining exactly what everything means in both film and video Edit Decision Lists. To the average person, an EDL is just a confusing list of numbers, so it is a neat trick to be able to pick one up and be able to explain it to everyone else in the room. The tape's presentation of EDLs is made clearer through a nice use of visuals.

I would have liked to see some more visual examples. For instance, rather than just seeing a list of the different film and video formats it would have been nice to see some actual tapes, and the output quality. It's one thing to be told that D1 video is the best digital tape format, it's another to see how it looks after six generations of dubs and to see how just how huge the tape is.

In Tape 4, *Pre-Visualization & Editing*, Forest Key looks at three different projects: two national car commercials and the short film. The commercials combine a mixture of techniques including 3D CGI, practical

models, and morphs. It is always interesting to see a project at different stages of development, and this tape shows these projects from the earliest stages of storyboards right through to the final cut. One interesting thing about the commercials is just how much was worked out in the pre-visualization (previz) process. On a commercial for Chevy, crude animation is shown as the initial concept. By the time we get to the final commercial, we see just how closely the piece stuck to that original animation, and how valuable even crude previz is to getting the right look and feel.

The highlight of the series for me was Forest going over the projects in the editing room and explaining things shot by shot (and in some cases, frame by frame). There was no specific detail about how certain elements were created, but that level of detail would have just mucked up the process. These tapes are all about getting out of the limited headspace of an animator and getting into the wider perspective of a Visual Effects Supervisor. This is the big picture.

Anyone who wants to get into visual effects, or any LightWave animator who wants to make the move to Visual Effects

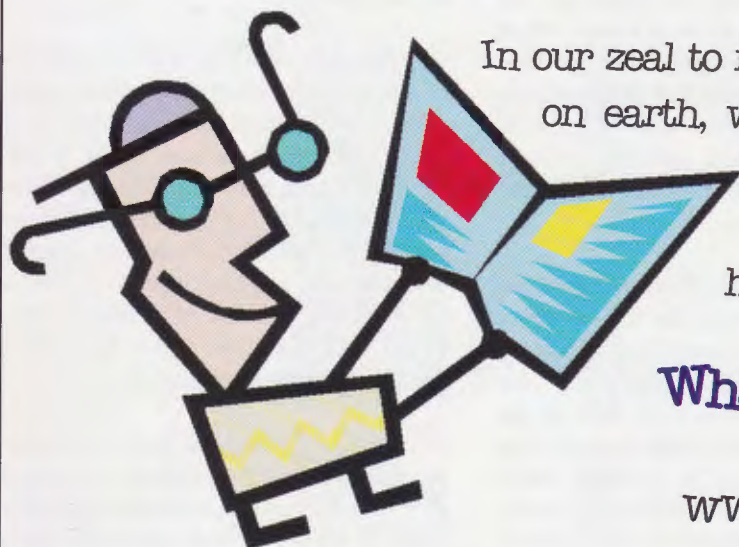
Supervisor, will be able to sound like a whiz kid if they get these tapes and absorb the contents. For the beginner they provide a solid foundation in the language and concepts of digital visual effects, and for the more experienced person they provide a nice way to fill in the inevitable gaps in self-taught knowledge. The series is off to a nice start, so keep watching for more from the *Masters of Visual Effects*.

The biggest issue for many people may be the price: are the tapes worth nearly three hundred bucks? After all, that would be a lot of books, magazine subscriptions, or plug-ins. However, I am aware of no other general visual effects video series, so in a sense this is the only game in town.

So, should you buy? It's a hard call for me to make in this case. These are tapes that give you general introductory information about the process of visual effects, but the price tag is not exactly introductory. On the other hand, this is knowledge that you will need to acquire one way or another. When you compare the price to the potential income increase you can get by moving up from animator to supervisor, it is probably a good investment for many people.

Attention LightWave Junkies

(and other assorted animator-types).



In our zeal to make Keyframe the coolest publication on earth, we print WAY too many issues every month. Needless to say, our warehouse is ready to burst! So we're having a **SALE!** on our back issues.

What Issues are You Missing?

www.keyframemag.com
888-778-WAVE

Continued from page 13

LightWave) and still frames that we loaded in as background images. Elements of the scene were updated in real time as elements on the set changed. Possibly the biggest benefit of this whole endeavor was our ability to get pertinent camera information (focal length, distance from the subject, etc.) right there on the set and check it against our scene. These things could have easily been overlooked or recorded incorrectly had we not been checking them in real time. This proved particularly important, as setting the LightWave camera to the focal length of the actual lens did not look correct. I believe we received incorrect information about the size of the CCD in the real camera. This affects the look of a shot created by a lens of a given focal length, and in this case, made it not match the LightWave scene. This would have been potentially problematic if we had merely recorded data on set and attempted the comp entirely in postproduction. Having the ability to check these parameters on the spot and modify our scene to suit the setup was fantastic.

I also took several photos with a digital still camera as a precaution in case we needed any additional information later. They proved unnecessary, but better safe than sorry. There was one thing that we did not do, which I would highly recommend. It would have been good to attempt to pull the matte on set to determine if any lighting anomalies could be corrected. We did run into this problem in post, but unfortunately it was too late to do anything. This was not a tragedy, it just required a bit more tweaking when pulling the matte from the bluescreen background. In this case, there was both a vertical background and a horizontal surface (the blue desktop) in front of the actor, Figure 2. The light striking the desk illuminated it much more than the background, making parts of it appear almost white! Is this the Fresnel effect in action?

Fortunately using software like Ultimatte allows you to use a screen correction layer, which can eliminate

such problems. If you are going to use Ultimatte or a similar tool, make sure you shoot a screen correction element. This is a shot of just the bluescreen with no actors. You will use it later to drop out any abnormalities in the screen. Screen correction can be very effective and is one of Ultimatte's powerful features. This can really help problem shots, but good lighting on a bluescreen is still very important to a composite's success. Having a good gaffer or DP with a lot of bluescreen experience is a great way to avoid these types of problems. If you have not shot bluescreen footage before and intend to spend some money renting time at a bluescreen studio, I strongly encourage you to invest in the added expense of an experienced professional gaffer to set up your lights.

Once our LightWave scene was setup and saved, our job on the set was done. We took the laptop back to the studio and got ready to do the composite. This was done with a combination of After Effects and LightWave. The BetaSP tape from the shoot was digitized using NewTek's Video Toaster NT, converted to .avi, and brought into After Effects, Figure 3. Unfortunately much of this could not happen on the Mac as the Video Toaster is currently available only for NT. Mac fans, this is your cue to demand NewTek make a Mac version of the Toaster. Email, call them, bug them at trade shows. I found myself thinking it would be a great idea for someone to make a plug-in that allows applications such as After Effects, Premiere, or Commotion to read .rtv files, which are native only to the Video Toaster NT.

In After Effects, an alpha channel was created from the bluescreen with the Ultimatte plug-in. This could have probably been done with the built-in chroma key tools in After Effects or Aura, but Ultimatte is the Cadillac of bluescreen software, and I like the results it provides. As with any project, there are many ways to achieve the same goal. This is just one of them, and it proved to be quite effective. Once the alpha channel was created for the thirty

second shot, each frame was output in an .iff sequence to facilitate importation into LightWave. I also converted the RGB data from our original bluescreen .avi footage into an .iff image sequence. We work with Alpha, Intel and Mac OS versions of LightWave and I find the native .iff format to be the most compatible and problem free. This becomes most apparent when dealing with ScreamerNet, really long image sequences, or generally, anything even mildly unusual.

The RGB sequence was loaded into the scene as a foreground image, with the alpha sequence masking out the blue background. The same sequence was mapped onto the desk with a high degree of transparency to create a subtle moving reflection as the actor spoke and moved his hands. The reflection is quite discrete, but adds a great deal of realism to the final animation, Figure 4. The results can be seen on this month's Keyframe CD. This technique can be easily applied to much more complex compositing shots and could really save a lot of headaches, time and money in the long run.

I hope this has been helpful and I welcome input from users who exercise the technique. By the way, the Irez card is available for the PC as well, so this is not a Mac-only technique!

Jason Goodman is the CEO of NYC based Multimedia Software Design. You can email him at jason@msd-nyc.com or visit their website, www.msd-nyc.com.



Figure 4: Final composite showing correct shadows and reflections.

Foundation Imaging's Workflow Pipeline for RSST



Hello again, and greetings from Valencia. It's core-of-the-Earth hot today, and I'm pausing every few minutes to pay tribute to the hard-working men and women of Southern California Edison who keep my air conditioner running. If it weren't for them and Sam Adam's Summer Ale, I don't know what I'd do. But I digress.

I finally met *Keyframe* editor Doug Kelly when he came to visit Foundation Imaging during SIGGRAPH. As I was showing him around the place, we talked about production and the workflow pipeline being used on *Starship Troopers* (now officially called *Roughnecks: The Starship Troopers Chronicles*, aka RSST). He thought it might make a good column, and I agreed.

Our current *Troopers* workflow model (yes, it's an evolutionary beast, always being refined or sometimes redeveloped - this week the animal gets opposable thumbs, maybe next week it loses the webbed feet - who knows?) was developed by Jeff Scheetz, our Project Lead on the show.

It starts like the others - a few weeks before animation starts on an episode, the modelers attack the scripts and storyboards looking for new designs. Modeling Supervisor Dave Adams has crews that specialize in organic modeling, and guys who excel at tech, and together they build stuff. A lot of stuff, actually. For a particularly complex model, a low-poly, often untextured 'stand-in' is built first, roughing out the model so that animators can begin using it in shots.

Character and bug hierarchies are handed off to Tom MacDougall and his team of animators, who are aces with the bones and the expressions and all of that fun stuff. They'll construct the many "Rico_On_Jetski.lws" or "Rippler_Bug_Attacking.lws" assembly files, and set up new characters or bugs with the tricked-out walk and run cycles.

By this point, Robert Baldwin and the technical directors have received terrain from the modelers, and are beginning to develop lighting for the episode. Each of the five-episode arcs in *Roughnecks* takes place on a different planet, so each one requires a different lighting style, but the lighting needs to be consistent across the scenes and episodes. More on that later.

So now we're getting into the animation. By the time animation begins, the CG director for that episode has watched an animatic or 'Leica reel' of his show and is getting ready to hand out the goods to his crew. For the record, we're running five animation teams concurrently:

Gold Team (Kevin Kipper), Red (PJ Foley), Blue (Dave Morton), Grey (Mike Stetson), and Green (Emile Smith). Each team is working on an episode, and their deliveries are offset slightly to avoid bottlenecks.

We work mostly with 'scenes' rather than individual shots for efficiency's sake; moving the camera within a scene can produce numerous shots without having to duplicate lighting or reprocess the same motion capture files. The first thing the animator is likely to do on receiving his or her scenes is book some time with Dave West and the Motion Capture gang. Each animator directs mocap for their scenes, the reason being that they are most familiar with the action.

While that mocap data is being processed, the animator will probably work on the facial animation if there is any dialog in the scene. Using a technique we call 'rotocap' the animator lip-syncs to the recorded dialog track, using a DPS PVR card and some custom software to grab a video file that is locked to the audio. This reference video file enables the animator to turn out a Morph Gizmo lip sync sequence fairly quickly.

So you've got terrain, you've got bugs, some motion-captured Troopers fighting them, and some dialog. Of course, there's still some animating to do. Mocap is handy, but it doesn't do everything for you. So the animator will take care of killing some bugs, animating some Troopers, adjusting the camera, and getting the performance of the scene right. The scene is rendered and the performance pass is edited into the Leica reel. The animator takes a deep breath at this point and, always the intrepid go-getter, dives headfirst back into the mosh pit for another scene.

The scene is now handed off to the TD team, where the lighting for that scene is polished to fit into bigger picture, and any needed effects are added (tracer fire, explosions, little tiny composited-in Buddy Lee action figures). Again, the shot is rendered and this final version dropped into the cut.

Before too long, you wind up with a show. By the time it gets to the client, our renderers will have generated over 28,000 frames of animation under the watchful eyes of Mike Donahue and the Systems department. The frames are recorded onto one of two DPS Hollywood machines, and a digital Betacam tape is written. "Razak's Roughnecks" finish another battle.

Steve Pugh is the General Manager at Foundation Imaging. You can contact him through their website, www.foundation-i.com.



Animator Marie Winnard directing motion capture performer Scott Altizer for a scene as motion capture specialist Emerson Johnson works the FILMBOX station. photo courtesy of Jared Bendis

A Natural Resource

HyperVoxels

Programming Plug-ins

Lightning

Surfacing

ScreamerNet

Storyboards and Maquettes

DPS Editing

Fresnel Shaders

Quick and Easy Trees

Smoke and Fire

Lens Flares

Aura

Business Tips

Modeling

Marketing Techniques

Natural Shaders

Steamer

Skyltracer

T2 Liquid Metal

Character Animation

Explosions

Macintosh

and more!

Get your backissues before they're gone!

888-778-WAVE www.keyframemag.com





LightWave® [6]

Allow yourself to be seduced.

Seduced by the power, the elegance and the flexibility. Your first encounter will reveal why LightWave dominates the demanding world of show business.

LightWave has always been the industry standard for exceptional realism. Today it just got better. Much better. Now, with caustics, radiosity and integrated Hypervoxels, LightWave [6] understands and delivers the nuances of lighting and volumetrics better than any other animation system on the market. The addition of a 96-bit floating-point render pipeline assures you ultimate color depth and accuracy.

LightWave [6] introduces the world's fastest and most accurate hybrid inverse/forward kinematics engine. Also, unique to LightWave [6] are IntelligEntities™. These include Endomorphs™ and Skelegons™ used to create and control character animation with efficiency and fluidity.

With the release of LightWave [6] nothing was left to chance. We completely redesigned the program and the process with you, the artist, in mind. The joy of LightWave, all the features, all the functions — better, faster, fun.

The seduction is complete.

NewTek booth #535 at Siggraph '99

www.newtek.com

Call (800) 862-7837

Outside the U.S. call (210) 370-8000

LightWave 3D is a registered trademark of NewTek, Inc. Hypervoxels, IntelligEntities, Endomorphs and Skelegons are trademarks of NewTek, Inc. © NewTek, 1999. All rights reserved.



NewTek

THE JOY OF SIX